Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Jernej Kernc

Orodje za interaktivno analizo časovnih vrst

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Blaž Zupan

Ljubljana, 2016

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Med pomembnejše tipe podatkov, ki jih danes srečamo na področju podatkovne analitike, sodijo časovne vrste. Kandidat naj preuči, kateri so tipični računski postopki in vizualizacije, s katerimi lahko analiziramo časovne vrste. Na podlagi te študije naj predlaga in razvije implementacijo teh orodij v izbranem programu za interaktivno podatkovno analitiko.

Izjava o avtorstvu zaključnega dela

Spodaj podpisani Jernej Kernc, študent z vpisno številko 63060111, avtor zaključnega dela z naslovom:

Orodje za interaktivno analizo časovnih vrst
(angl. *Toolbox for interactive time series analysis*)

IZJAVLJAM

1. da sem pisno zaključno delo študija izdelal samostojno pod mentorstvom prof. dr. Blaža Zupana;
2. da je tiskana oblika pisnega zaključnega dela študija istovetna elektronski;
3. da sem pridobil vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v pisnem zaključnem delu študija in jih v delu jasno označil;
4. da sem pri pripravi pisnega zaključnega dela študija ravnal v skladu z etičnimi načeli in, kjer potrebno, za raziskavo pridobil soglasje etične komisije;
5. soglašam, da se elektronska oblika pisnega zaključnega dela študija uporabi za preverjanje podobnosti vsebine z drugimi deli;
6. da na Univerzo v Ljubljani neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve avtorskega dela v elektronski obliki, pravico reproduciranja ter pravico dajanja dela na voljo javnosti na svetovnem spletu;
7. dovoljujem objavo svojih osebnih podatkov, ki so navedeni v pisnem zaključnem delu študija in tej izjavi, skupaj z objavo pisnega zaključnega dela študija.

Ljubljana, 6. julij 2016 JERNEJ KERNC

Hvala bogu!

Kazalo

Povzetek

Časovne vrste, kakor pravimo primerom zaporedja meritev opazovanega pojava, predstavljajo pomemben tip podatkov v ekonometriki (npr. gibanje letnega BDP in relativne zadolženosti držav), v poslovnem svetu (npr. prodajna uspešnost produkta po mesecih), v medicini (EEG, EKG), v meteorologiji (npr. sprememba povprečne temperature skozi čas) in na skoraj skoraj vseh ostalih področjih naravoslovnih in družbenih ved. Pomembno je, da imamo na voljo orodja, s katerimi lahko časovne vrste ustrezno proučujemo, transformiramo, analiziramo, vizualiziramo in modeliramo. V diplomskem delu smo, temelječ na programskem paketu za podatkovno rudarjenje Orange, razvili odprtokodno orodje za interaktivno analizo, vizualizacijo in napovedovanje časovnih vrst. Razširitev obsega štirinajst gradnikov podatkovnih tokov v smislu vizualnega programiranja, s katerimi je mogoče časovne vrste odvajati, interpolirati, agregirati, sezonsko prilagoditi, transformirati z okenskimi transformacijami in ocenjevati kavzalnost med vrstami. Razvili smo tudi komponente za prikaz časovnih vrst v črtnem diagramu, periodogramu, korelogramu in v spiralni toplotni karti. Za modeliranje smo v knjižnico vključili napovedna modela VAR in ARIMA. Izdelek smo preizkusili in ovrednotili na različnih naborih podatkov.

Ključne besede: časovne vrste, vizualizacija, avtoregresija, avtokorelacija, ARIMA, VAR, napovedovanje, strojno učenje, podatkovno rudarjenje, vizualno programiranje, umetna inteligenca, Orange

Abstract

Time series, as we call sequences of measurements of an observed phenomenon, represent an important type of data in the fields of econometrics (e.g. countries' yearly GDP and relative debt change), business (e.g. number of products sold per month), medicine (EEG, ECG), meteorology (e.g. change in average temperature through time) and in almost all other fields of natural and social science. It is thus important for toolsets to exist, with which one can analyze, transform, visualize, and model time series data. Based on renowned Orange data mining software framework, we propose a suite of visual programming widgets for construction of workflows for interactive time series analysis, visualization, and forecast. In particular, the suite comprises widgets for time series differencing, interpolation, aggregation, seasonal adjustment, transformation with window functions and estimation of causality. Additionally, we devise components for plotting time series data in a line chart diagram, periodogram, correlogram, and spiral heatmap. We support time series modeling with VAR or ARIMA models. We evaluate our contribution on various time series data sets.

Keywords: time series, visualization, autoregression, autocorrelation, ARIMA, VAR, forecast, machine learning, data mining, visual programming, artificial intelligence, Orange

Poglavje 1

Uvod

"Time discovers truth."

Seneca

Danes večina zbirk podatkov nastaja inkrementalno — novi vnosi se beležijo, kot se porajajo dogodki opazovanega pojava. Kronološko urejenim zaporedjem števil, ki izražajo spremembo pojava v času, pravimo *časovne vrste*. Še tako raznovrstne in kompleksne podatke je, če odražajo različna časovna obdobja, mogoče projicirati na časovno dimenzijo in dimenzijo vrednosti enega izbranega opazovanega pojava. Ustrezni primeri časovnih vrst so, na primer, vrednost delnice v zaporednih dneh, dobiček podjetja v zaporednih letih, število prodanih artiklov v zaporednih tednih, mesecih ali letih; dnevna izmerjena temperatura zraka v nekem mestu, količina vodnega pretoka neke reke po dnevih, mesečna višina podtalnice na nekem območju, količina proizvedenih odpadkov v neki državi po letih, število zaklanih prašičev v razvitem svetu po letih, EEG in EKG v medicini ter PCM zvočni zapis.

Časovne vrste nam dajejo vpogled v *dinamiko* pojava in s tem, upajmo, v zakonitosti, ki ga porajajo. Analizo in metode napovedovanja časovnih vrst s pridom uporabljamo na področjih, kot so poslovni svet in ekonomika, ekonometrika, medicina, meteorologija, oceanografija, digitalno procesiranje signalov in na skoraj vseh ostalih področjih naravoslovnih znanosti in družbenih ved. Ena prvih proučevanih časovnih vrst je vrsta mesečnega števila pojavitev sončevih peg v astronomiji [32]. Danes pa je precej aktualno proučevanje geoklimatskih sprememb, makroekonomskih kazalcev in napovedovanje valutnih križev.

Za analizo časovnih vrst imamo danes na razpolago že množico orodij[1]. Če naštejemo samo nekaj bolj znanih med njimi: R, Python Pandas, SageMath, MathWorks Matlab,

[1] `https://en.wikipedia.org/wiki/Comparison_of_statistical_packages#Time_series_analysis`

Microsoft Excel, IBM SPSS, Stata, Wolfram Mathematica, SAS. Vsem naštetim orodjem, pa tudi večini neomenjenih, lahko pripišemo eno ali celo obe naslednji pomanjkljivosti:

- *Orodja niso prosto dostopna.* Zajetne licenčnine in omejujoči pogoji uporabe lahko naredijo sicer tehnično-dovršeno orodje popolnoma neuporabno, če si ga večina potencialnih uporabnikov ne more privoščiti. Dodatno je pri zaprti programski opremi oteženo vplivanje na nadaljnji razvoj, nemogoče pa je tudi zanesljivo preveriti pravilnost algoritmov.

- *Orodja so zapletena za uporabo.* Poleg domenskega znanja, ki naj bi ga uporabnik pridobil drugje, zahteva uporaba večine omenjenih orodij tudi poznavanje vsaj enega programskega jezika, principov programiranja, obdelave podatkov in druga tehnična ter matematična znanja.

V okviru tega diplomskega dela smo razvili prosto dostopno programsko rešitev za analizo, predobdelavo, vizualizacijo in napovedovanje prihodnjih vrednosti časovnih vrst. Rešitev uporablja interaktiven uporabniški vmesnik, ki daje uporabniku takojšnjo ali skoraj takojšnjo povratno informacijo o statusu trenutnega izračuna ali celo kar o rezultatih. Vmesnik temelji na principih vizualnega programiranje in je zatorej, v okviru vseh svojih omejitev, precej preprost za uporabo.

V poglavju 2 formaliziramo pojem časovne vrste, opišemo dva modela dekompozicije časovnih vrst na komponente in opredelimo nekaj lastnosti časovnih vrst, ki lahko pomembno vplivajo na izbor nadaljnjih korakov obravnave. V poglavju 3 se posvetimo nekaterim načinom modeliranja časovnih vrst. Podrobneje predstavimo dva tipa modelov, VAR in ARIMA, ki sta pogosto uporabljena za napovedovanje prihodnjih vrednosti časovnih vrst. Orišemo tudi način vrednotenja modelov in definiramo smiselne mere napake, ki nam lahko pomagajo izbrati najbolj ustrezen model. V poglavju 4 predstavimo naš glavni prispevek, razširitev za programski paket Orange, ki omogoča interaktivno analizo, vizualizacijo in napovedovanje časovnih vrst s preprostim grajenjem podatkovnega toka po principih vizualnega programiranja. Razširitev obsega štirinajst v podpoglavjih podrobneje opisanih gradnikov, s katerimi je mogoče časovne vrste najmanj obdelovati, prikazovati, kopičiti, razstavljati in z modeli napovedovati. V poglavju 5 prikažemo primer uporabe rešitve za napoved dolgoročnega globalnega temperaturnega trenda. V poglavju 6 na kratko ovrednotimo naš prispevek ter podamo nekaj konkretnih predlogov za nadaljnje delo in izboljšanje izdelane programske rešitve. Tu podamo tudi nekaj bolj splošnih raziskovalnih usmeritev v področja in postopke, ki naj bi jih polno-funkcionalno splošno-namensko orodje za analizo časovnih vrst eventualno podpiralo.

Poglavje 2

Časovni podatki in časovne vrste

"How long a minute is depends on which side of the bathroom door you're on."

Zall's Second Law

Podatkom, ki vsebujejo komponento časa ali se s časom spreminjajo, pravimo časovni podatki (angl. *temporal data*). Časovne podatke lahko razdelimo na [27]:

- *dogodkovne podatke*, ki imajo prirejen čas nastopa dogodka,
- *intervalne podatke*, ki imajo definiran začetek in konec,
- *sekvence* ali zaporedja, npr. zaporedje obiskanih spletnih strani pred nakupom, in
- *časovne vrste*, ki predstavljajo zaporedne meritve ene same zvezne spremenljivke.

2.1 Časovne vrste

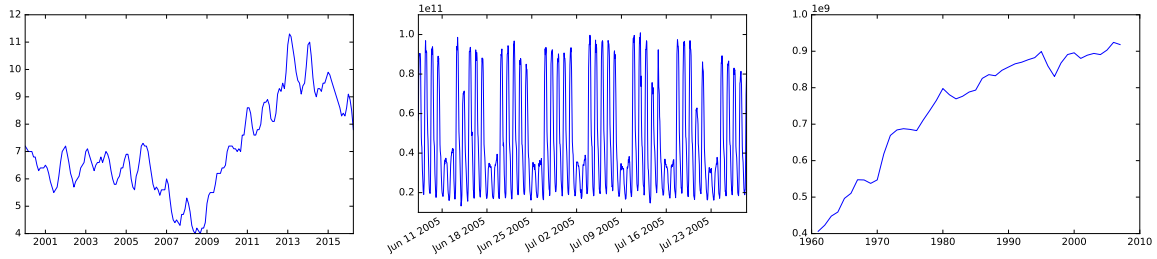
Časovna vrsta (angl. *time series*) je zaporedje meritev ene spremenljivke v času. Univariatno časovno vrsto, ki je odvisna od le ene vhodne, časovne spremenljivke, lahko simbolno opišemo z enačbo:

$$y_i = f(t_i) \tag{2.1}$$

multivariatno časovno vrsto, na katero sočasno vpliva več spremenljivk, pa kot:

$$y_i = f(t_i, \mathbf{X}) \tag{2.2}$$

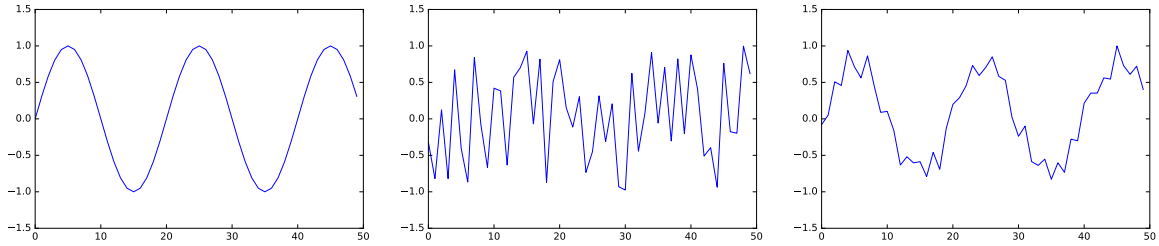
kjer y_i predstavlja izmerjeno vrednost opazovane spremenljivke v času t_i. Nekaj primerov časovnih vrst je na sliki 2.1.



(a) (b) (c)

Slika 2.1: Primeri časovnih vrst: (a) raven nezaposlenosti (v %) v Sloveniji po mesecih v letih pred in po gospodarski krizi (vir: Eurostat); (b) čezatlantski internetni pretok (v bitih) večjega evropskega ponudnika dostopa do interneta [5]; (c) število gojenih prašičev na svetu (v glavah) (vir: Organizacija Združenih narodov za prehrano in kmetijstvo).

Časovne vrste so lahko posledica deterministčnih procesov, kjer je mogoča izpeljava prihodnjih stanj iz preteklih na deterministčen, vnaprej določen način, ali pa so posledica stohastičnih procesov, kjer na prihodnja stanja vpliva (tudi) neka zunanja, neopazovana, naključna spremenljivka. V praksi so časovne vrste pogosto kombinacija deterministčnih komponent (trend, preskok, periodičnost) nadgrajenih s stohastičnimi, ki vplivajo na spremembe v manjšem obsegu in pri višjih ločljivostih [24]. Primeri časovnih vrst, odvisnih od narave procesov, ki jih proizvajajo, so prikazani na sliki 2.2.



(a) (b) (c)

Slika 2.2: (a) sinusoida, kot jo poraja deterministčni proces $y(t) = \sin(\frac{2\pi}{20} \cdot t)$; (b) popolnoma stohastična časovna vrsta; (c) časovna vrsta, ki je kombinacija deterministčnega in stohastičnega procesa.

2.2 Dekompozicija časovnih vrst

Ne glede na resnični izvorni proces lahko časovne vrste razstavimo na naslednje komponente [15]:

- *Trend*. Predstavlja dolgoročno napredovanje (višanje ali nižanje povprečnih vrednosti) časovne vrste. Lahko se spreminja. Ni nujno, da je linearen.

- *Sezonska komponenta*. Obstaja, kadar ima vrsta značilen sezonski vpliv (npr. število turistov je na severni polobli večje v poletnih mesecih). Dolžina sezone je fiksna in vnaprej znana.

- *Ciklična komponenta*. Obstaja, kadar je za vrsto značilna cikličnost (vzponov in padcev), ki pa ni nujno fiksne dolžine. Tudi magnituda ciklične komponente je navadno bolj spremenljiva kot magnituda sezonske komponente. Dolžina periode ciklov je običajno vsaj dve leti.

- *Ostanek* (angl. *residual*). Je iregularna, naključna, šumna komponenta. Tudi napaka. Predstavlja tisto, kar od vrste ostane, če ji odvzamemo vse ostale komponente.

Časovno vrsto lahko na zgoraj navedene komponente razgradimo na podlagi aditivnega modela:

$$y_t = T_t + S_t + C_t + \varepsilon_t \tag{2.3}$$

ali na podlagi multiplikativnega modela:

$$y_t = T_t \cdot S_t \cdot C_t \cdot \varepsilon_t \tag{2.4}$$

kjer y_t predstavlja vrednost vrste v času t; T_t, S_t, C_t, in ε_t pa so vrednosti komponente trenda, sezonske komponente, ciklične komponente in ostanka v času t. Nekateri avtorji navajajo tudi razdelitev vrste na dodatno odvisno komponento stohastičnega procesa [24, 27] ali pa razdelitev na enotno *trend-cikel* komponento [15]. Primer dekompozicije časovne vrste je prikazan na sliki 4.4.

Aditivni model razgradnje (enačba 2.3) je primeren, kadar magnituda sezonskih nihanj ni odvisna od vrednosti časovne vrste. Če pa višje vrednosti časovne vrste sovpadajo z večjimi odkloni v visoki sezoni (ali obratno), potem je bolj primeren multiplikativni model (enačba 2.4).

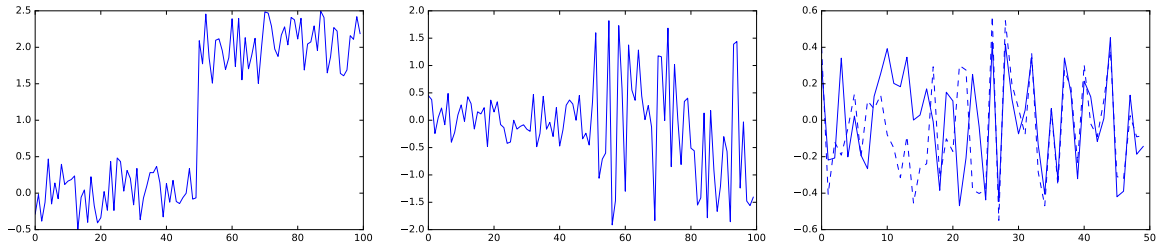
Časovnim vrstam z odstranjeno sezonsko komponento, tj. $y_t - S_t$ oz. $\frac{y_t}{S_t}$, pravimo *sezonsko prilagojene* (angl. *seasonally adjusted*) in jih pogosto srečamo v uradnih statistikah. Kadar napovedni model učimo na sezonsko prilagojeni časovni vrsti (včasih je zaželeno zgraditi model na vsaki komponenti posebej [15]), pridobljenim napovedim ne smemo pozabiti vrniti sezonske komponente (oz. moramo napovedi posameznih komponent nazaj združiti).

2.3 Lastnosti časovnih vrst

Veliko postopkov in algoritmov za delo s časovnimi vrstami predpostavlja specifične vrednosti nekaterih lastnosti časovnih vrst, ki jih sicer lahko ocenimo iz podatkov.

2.3.1 Stacionarnost

Večina algoritmov za analizo in napoved časovnih vrst predpostavlja, da so časovne vrste *stacionarne* (angl. *stationary*), kar pomeni, da se njihovo povprečje, varianca ter avtokorelacija s časom ne spreminjajo [27]. Primeri teh sprememb so prikazani na sliki 2.3.



(a) (b) (c)

Slika 2.3: Sprememba (a) povprečne vrednosti, (b) variance in (c) korelacije. Če časovna vrsta vsebuje katero od teh sprememb, pravimo, da ni stacionarna.

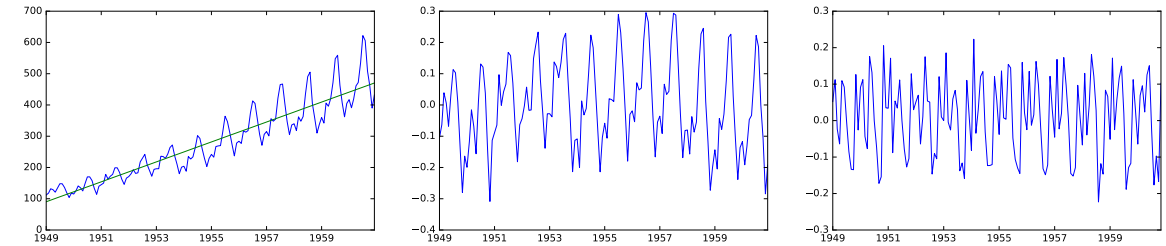
Različno porazdeljeno varianco rešujemo z logaritmiranjem vrednosti časovne vrste, spreminjajoče povprečje (trend) pa z deljenjem z ustrezno drugo znano časovno vrsto (npr. število nakupov vozil delimo s kupno močjo v vsakem obdobju). Priljubljena metoda za reševanje problema povprečja in avtokorelacije, ki nista konstantna, je numerično odvajanje (angl. *differencing*):

$$y'_t = y_t - y_{t-1} \tag{2.5}$$

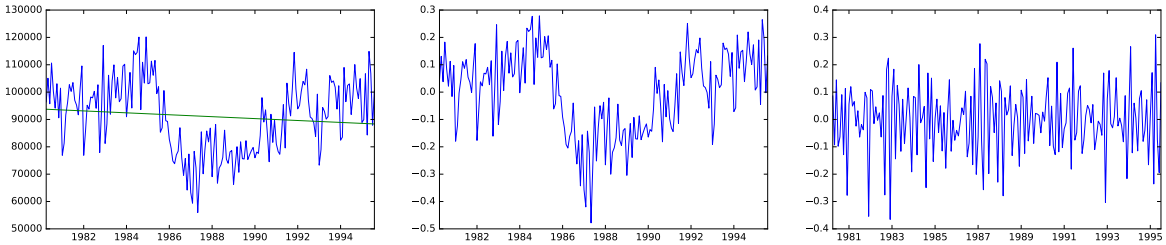
Z modeliranjem in analiziranjem tako modificirane časovne vrste torej v resnici modeliramo in analiziramo *spremembe* v originalni vrsti.

Slika 2.4 prikazuje dve časovni vrsti, ki nista stacionarni, saj imata očiten linearni trend oz. odsekoma različno povprečno vrednost in različen odklon glede na obdobje. Poleg njiju sta transformirani časovni vrsti z bolj ali manj ohranjenimi lastnostmi. Skrajno desna prikaza sta stacionarna.

Ena od metod zaznavanja sprememb v časovni vrsti je primerjanje lastnosti porazdelitev vrednosti v dveh različnih časovnih oknih [9, 18]. Če postaneta porazdelitvi v oknih v nekem trenutku bistveno različni, veljajo za vrsto odtlej drugačna pravila in je potrebno



(a)



(b)

Slika 2.4: Prikaz od leve proti desni: (a) časovne vrste mesečnega števila potnikov mednarodnega letalskega prometa, v tisočih [3], logaritmirane časovne vrste z odštetim linearnim trendom in numeričnega odvoda logaritmirane časovne vrste; (b) število mesečno zaklanih prašičev v avstralski zvezni državi Victoria (vir: Avstralski statistični urad).

naučene modele ponastaviti.

V ekonometriki se za oceno stacionarnosti oz. potrebe po odvajanju časovne vrste navadno uporabi statistične teste za testiranje prisotnosti enotskega korena (angl. *unit root*) v karakteristični enačbi, kot so npr. razširjeni Dickey-Fullerjev test (ADF), Phillips-Perronov test in test Kwiatkowski-Phillips-Schmidt-Shin (KPSS) [15, 25].

2.3.2 Enakomernost razmika

Pomembna lastnost časovnih vrst je, da meritve nastopajo v *enakomernih razmikih* (angl. *equispaced*) [27], torej, z uporabo t_i iz enačbe 2.1, velja, da je časovna razlika:

$$\Delta t = t_i - t_{i-1} \tag{2.6}$$

med vsemi meritvami konstantna.

Večina modelov, ki za napoved vključujejo pretekle vrednosti časovne vrste, zasnovanih s primitivi linearne algebre, se na pretekle vrednosti vrste opira le simbolno, npr. y_{t-1}, y_{t-2} ..., tj. z implicitno razdaljo med meritvami enako ena (enota). Neenakomerno razmaknjene meritve v časovni vrsti lahko torej predstavljajo problem. Rešujemo ga tako, da vmesne manjkajoče vrednosti, če je teh relativno malo, interpoliramo, ali pa signal

vnovič vzorčimo pri nižji frekvenci, kjer sosednje meritve združujemo po vnaprej določeni agregatni funkciji. Na primer, časovno vrsto osebnega dnevnega vnosa beljakovin, kjer nam vsak teden podatek za kak dan manjka, pretvorimo v časovno vrsto tedenskega vnosa, kjer meritve posameznega tedna seštejemo.

2.3.3 Periodičnost

Časovno vrsto lahko sestavlja ena ali več periodičnih komponent. To so komponente, ki se v vrsti s konstantno ali približno konstantno periodo ponavljajo. Sezonska komponenta, podrobneje opisana v razdelku 2.2, je ena takih komponent. Kadar ima časovna vrsta izrazite periodične komponente, je te mogoče videti v prikazu *gostote močnostnega spektra* (angl. *power spectral density, PSD*), periodogramu [32]. Primer periodograma je prikazan na sliki 4.6a. Gostoto močnostnega spektra signala je mogoče izračunati z diskretno Fourierovo transformacijo (DFT), učinkovito z algoritmom FFT.

2.3.4 Manjkajoči podatki in šum

Pogosto se zgodi, da v časovnih vrstah nastopajo napake v obliki *manjkajočih podatkov* (angl. *missing data*), kjer vrednosti meritev ob določenih časih niso podane, ali v obliki *šuma* (angl. *noise*), kar pomeni, da so izmerjene vrednosti nenatančne. Vzroki za napake so lahko različni: nedelujoča oprema, človeški faktor, okoljske razmere.

V statistiki se nadomeščanju neznanih vrednosti reče imputacija (angl. *imputation*), pri časovnih vrstah pa je najpogostejši postopek imputacije *interpolacija*, Če je manjkajočih vrednosti v vrsti relativno malo, lahko te ocenimo z interpolacijo. Popularni metodi sta linearna interpolacija in interpolacija s kubičnimi polinomskimi zlepki (angl. *spline interpolation*) [27]. V primeru, da je manjkajočih podatkov veliko, je, kadar je to mogoče, navadno bolje, da teh segmentov vrste v modelih ne upoštevamo, saj interpolacija sama predstavlja primitivni napovedni model in kot taka v podatke vnese svoj šum in pristranosti (angl. *bias*) [31]. Napako šuma odpravljamo s *porazdeljevanjem* (angl. *binning*), kjer vsaki vrednosti v časovni vrsti pripišemo najbližjo vrednost iz vnaprej izbrane porazdelitve, ali pa z *glajenjem* (angl. *smoothing*) z drsečim oknom (angl. *sliding window*) [27].

Poglavje 3

Modeliranje in analiza časovnih vrst

"It is far better to foresee even without certainty than not to foresee at all."

Henri Poincare

Cilj metod strojnega učenja je izgradnja dobrega opisnega in napovednega modela. Kar zadeva časovne vrste, želimo dober napovedni model, ki bo bodisi zanesljivo *ekstrapoliral* prihodnje vrednosti vrste, npr. da bomo z minimalnim tveganjem obogateli na delniških trgih, bodisi bo vrsto glede na lastnosti ustrezno *označil* oz. vrste med seboj *razlikoval*, npr. EKG sliki zdravega in srčno-bolnega človeka.

Za reševanje obeh nalog obstajajo različni prijemi, tako v frekvenčni kot v časovni domeni. Analiza časovnih vrst v frekvenčni domeni ima največji potencial pri delu z zvokom in s slikami, v elektrotehniki, pri digitalnem procesiranju signalov, kjer je signal navadno opisan s kombinacijo osnovnih frekvenc in njihovih višjih harmonikov. Za časovne vrste, kot jih srečamo v ekonometriki in v poslovnem svetu, kjer na gibanje vrste navadno ne vplivajo dejavniki s fiksno periodično naravo oz. so periode predolge, da bi jih zanesljivo zaznali, pa je najbrž bolj primerna analiza v časovni domeni. Tej se v nadaljevanju tudi posvečamo v tem delu.

3.1 Klasifikacija in razvrščanje

Ali nek EKG zapis, časovna vrsta meritev električne depolarizacije človeškega srca pri bitju, ustreza popolnoma zdravemu človeku ali pa morda nakazuje srčno napako? Pomembno je, da znamo časovne vrste med seboj razlikovati, jih ustrezno klasificirati oz. razvrstiti v smiselne gruče.

Večina ustaljenih metod za klasifikacijo in razvrščanje (angl. *clustering*) časovnih vrst se opira na mero razdalje med dvema vrstama. Pogosto uporabljeni meri razdalje sta evklidska razdalja in razdalja DTW.

Prednost evklidske razdalje je, da jo je mogoče izračunati v linearnem času, slabost pa, da morata biti obe vrsti enake dolžine in da že najmanjši zamik dveh sicer izjemno podobnih vrst (npr. $\{y_t, \ldots, y_{t+m}\}$ in korak zamaknjene $\{y_{t+1}, \ldots, y_{t+1+m}\}$) povzroči, da je dobljena razdalja lahko nesorazmerno velika.

Bolj uporabna je razdalja izračunana z dinamičnim časovnim prilagajanjem (angl. *digital time warping*, DTW). Algoritem je v splošnem bolj kompleksen (časovna kompleksnost $O(n^2)$), a obstajajo hitrejše ($O(n)$) implementacije, npr. FastDTW. Med bolj uspešnimi algoritmi za klasifikacijo časovnih vrst je 1-NN DTW [38]. Gre za metodo iskanja k najbližjih sosedov (angl. *k-nearest neighbors, k-NN*), kjer je $k = 1$, razdalja med različnimi vrstami pa je razdalja DTW. Napovedani razred je enak razredu najbližjega soseda.

S klasifikacijo in razvrščanjem časovnih vrst se v tem delu ne bomo podrobneje ukvarjali. Nekaj predlogov za nadaljevanje raziskovanja podamo v razdelku 6.1.

3.2 Napoved nadaljevanja časovne vrste

V analizi časovnih vrst pomembno vlogo igra napovedovanje prihodnjih vrednosti (angl. *forecasting*). Kadar imamo opravka s časovnimi vrstami večih, povezanih spremenljivk, so za izgradnjo napovednih modelov povsem legitimna izbira klasične metode strojnega učenja: regresijske metode, drevesa, naivni Bayes, SVM, nevronske mreže, itd. [9, 27] Tako predpostavimo, da so posamezne meritve (instance, vrstice) neodvisne, časovna os pa predstavlja samo dodaten stolpec v atributnem prostoru.

Glede na naravo časovnih vrst, da si zaporedne vrednosti sledijo in torej meritve med seboj *niso* neodvisne (visoke sledijo visokim in nizke nizkim), pa so lahko bolj zanimive in uspešne prilagojene metode. En tako prilagojenih bolj popularnih modelov v ekonometriki [34] je *model vektorske avtoregresije* ali *model VAR*. S simetrično obravnavo več spremenljivk predstavlja generalizacijo univariatnega avtoregresijskega modela. Vhod v model je stopnja zamika p in seznam k predvidoma povezanih spremenljivk izmerjenih v časih $t = 1, \ldots, T$. Model VAR(p) predstavlja enačba [41]:

$$\mathbf{Y}_t = \mathbf{c} + \sum_{i=1}^{p} \mathbf{\Phi}_i \mathbf{Y}_{t-i} + \boldsymbol{\varepsilon}_t \tag{3.1}$$

kjer je $\mathbf{Y}_t$ $(k \times 1)$ vektor vrednosti spremenljivk v času t, $\mathbf{c}$ $(k \times 1)$ vektor konstant, $\mathbf{\Phi}_i$

$(k \times k)$ matrika koeficientov avtoregresijskega vpliva za i korakov zamaknjenih meritev, $\boldsymbol{\varepsilon}_t$ pa je $(k \times 1)$ vektor naključnih, neodvisnih in identično porazdeljenih (angl. *independent and identically distributed, i.i.d.*) napak.

Podobno popularen je *model ARMA* [3], ki poleg p avtoregresijskih členov vsebuje tudi vrednosti napak preteklih q napovedi (iz angl. *moving average, MA model*). Model ARMA(p, d, q) opišemo z enačbo:

$$y_t = c + \underbrace{\sum_{i=1}^{p} \phi_i y_{t-i}}_{\text{členi AR}} + \underbrace{\sum_{i=1}^{q} \theta_i \varepsilon_{t-i}}_{\text{členi MA}} + \varepsilon_t \tag{3.2}$$

kjer je y_t vrednost časovne vrste v času t, c konstanta, ϕ_i ter θ_i koeficienti členov AR ter MA modelov, ε_t pa so naključne, nezajete napake (ponovno, i.i.d., Gaussov beli šum). Model ARMA je od začetka svoje popularizacije v 1970-ih doživel nekaj razširitev oz. generalizacij [9]:

- *ARIMA*(p, d, q), ki poleg AR in MA parametrov vsebuje še integracijski parameter d. Časovna vrsta je pred prilagajanjem modela numerično odvedena s stopnjo d (glej razdelek 2.3.1), pri napovedi pa je potrebno vrednosti nazaj integrirati.

- SARIMA$(p, d, q)(P, D, Q)_m$ ali sezonska ARIMA, ki ima dodatne parametre za modeliranje sezonskega gibanja vrste;

- ARMAX, ki v izračunu vrste Y upošteva tudi nabor zunanjih (angl. *exogenous*) dejavnikov;

- NARMA ali nelinearna ARMA, ki dopušča, da so AR in MA členi v nelinearni odvisnosti;

- VARMA, ki razširja avtoregresijo na več spremenljivk; idr.

Modele, ki vsebujejo samo člene AR (npr. VAR, ARIMA$(2, 1, 0)$), je mogoče enostavno prilagoditi in njihove koeficiente učinkovito izračunati po metodi najmanjših kvadratov (angl. *ordinary least squares regression, OLS*). Modele, ki vsebujejo tudi člene MA (npr. ARIMA$(0, 1, 2)$) pa prilagajamo z metodo največjega verjetja (angl. *maximum likelihood estimation, MLE*), ki maksimira skupno gostoto verjetnosti preko vseh koeficientov [3] in z rekurzivnimi metodami [12].

3.2.1 Določanje števila členov AR in MA

Kako določimo število avtoregresijskih členov (p), členov napake (q) ali stopenj odvajanja (d), ki naj nastopajo v našem modelu? Obstajajo izkustvene ocene, ki temeljijo na obliki avtokorelacijske (ACF) in delne avtokorelacijske funkcije (PACF) [28, 29]. Primerna stopnja modela je zadnja perioda, ki ima v ACF ali PACF znatno (statistično pomembno) vrednost. Nekaj izkustvenih pravil je zajetih v tabeli 3.1.

Oblika ACF / PACF	**Ustrezen model**
ACF eksponentno pada ali sinusno konvergira k ničli.	AR model; red (parameter p) določa zadnja pomembna perioda PACF.
ACF ima enega ali več znatnih vrhov ali pa PACF eksponentno ali sinusno konvergira k ničli.	MA model; red enak periodi, kjer se v ACF začnejo ničelne vrednosti.
ACF konvergenca k nič, ki se začne po nekaj periodah.	ARMA model.
Vse vrednosti ACF/PACF okrog nič.	Ni ustreznega modela; podatki so naključni.
ACF ima visoke vrednosti v fiksnem intervalu.	Časovna vrsta vsebuje sezonsko komponento.
ACF vrednosti ne konvergirajo k nič ali pa je znatnih period veliko.	Časovna vrsta ni stacionarna, potrebna je transformacija (odvajanje). Model ARIMA.
ACF vrednost pri zamiku 1 je znatno negativna.	Vrsta je bila odvedena prevečkrat.

Tabela 3.1: Izbira modela in parametrov (p, d, q) v odvisnosti od oblike ACF ali PACF.

Alternativne, informacijsko-teoretične ocene za primernost modelov so razni informacijski kriteriji, npr. FPE, AIC, AIC_c ali BIC [4, 15, 23], ki iščejo optimalno ravnovesje med točnostjo prileganja in številom parametrov (stopnjo p, q) modela.

3.2.2 Vrednotenje napovednih modelov in mere napake

V strojnem učenju se za oceno pričakovane pravilnosti napovedi modelov uporablja *k-kratno prečno preverjanje* (angl. *k-fold cross validation*). Za časovne vrste pa ta način vrednotenja modelov ni najbolj primeren, saj podatke za učno in testno množico pri prečnem preverjanju vzorčimo *naključno*, v časovnih vrstah, kjer so prihodnje vrednosti odvisne tudi od prejšnjih, pa je pomembno, da model učimo na seriji *zaporednih* podatkov.

Ustaljena metoda za prečno preverjanje modelov časovnih vrst je ta, da na delu znanih podatkov $\{y_1, \ldots, y_t\}$ modele učimo, ocenjujemo pa jih na delu izven učnega vzorca $\{y_{t+1}, \ldots, y_{t+n}\}$ (angl. *out-of-sample, OOS*) [2]. Ocenjujemo lahko uspešnost napovedi za $n \geq 1$ prihodnjih korakov.

Izmerimo lahko tudi aproksimacijsko napako, ki jo model stori na učnih (angl. *in-sample*) podatkih. Če model dobro zajame odvisnosti v učnih podatkih, potem v ostankih (angl. *residuals*):

$$\varepsilon_t = y_t - \hat{y}_t \tag{3.3}$$

ni več nobenih odvisnosti (ACF in PACF imata vse vrednosti blizu nič) – izgledajo kot i.i.d. šum [9, 15].

Kako vemo, ali neka OOS napaka implicira dober, zanesljiv model? Primerjamo jo lahko z napovedno napako storjeno na vzorčenih, učnih podatkih. Če se napaki ne razlikujeta preveč, je model stabilen in OOS napaka predstavlja relativno dobro oceno realne napake storjene v napovedi [15].

Nekatere pogosto uporabljene mere napake za oceno primernosti modela so [27]:

- koren povprečne kvadratne napake (angl. *root-mean-square error, RMSE*):

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n} (y_t - \hat{y}_t)^2}{n}} \tag{3.4}$$

- povprečna absolutna napaka (angl. *mean absolute error, MAE*):

$$MAE = \frac{1}{n} \sum_{t=1}^{n} |y_t - \hat{y}_t| \tag{3.5}$$

- povprečna absolutna procentna napaka (angl. *mean absolute percent error, MAPE*):

$$MAPE = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{y_t - \hat{y}_t}{\overline{y}} \right| \tag{3.6}$$

- napoved spremembe smeri (angl. *prediction of change in direction, POCID*):

$$POCID = \frac{100}{n} \sum_{t=2}^{n} |(y_t - y_{t-1})(\hat{y}_t - \hat{y}_{t-1}) > 0| \tag{3.7}$$

- koeficient določnosti (angl. *coefficient of determination*):

$$R^2 = 1 - \frac{\sum_{t=1}^{n} (y_t - \hat{y}_t)^2}{\sum_{t=1}^{n} (y_t - \overline{y})^2} \tag{3.8}$$

kjer so y_t znane resnične vrednosti časovne vrste, $\overline{y}$ povprečna vrednost časovne vrste, $\hat{y}_t$ pa napovedi.

Mera POCID podaja oceno, kolikšen procent napovedi pravilno predvidi smer spremembe (tj. navzgor ali navzdol), mera R^2 pa pove, kolikšen delež variance v časovni vrsti pojasni napovedni model. Poudariti velja, da meri POCID in R^2 nista napaki, temveč sta indeksa, kjer višje vrednosti pomenijo boljši napovedni model.

Prednost napake MAPE je, da je neodvisna od povprečne vrednosti časovne vrste, kar jo naredi bolj primerno za medsebojno primerjavo različnih modelov (različnih časovnih vrst) [28]. Njena slabost je, da je nestabilna pri $\overline{y} \approx 0$.

3.3 Grangerjeva kavzalnost

Grangerjeva kavzalnost (vzročnost) je statistični test, ki ga izvedemo nad parom časovnih vrst in ki nam pove, ali je ena časovna vrsta lahko uporabna pri napovedovanju druge [10].

Test prilagodi avtoregresijski model dveh *stacionarnih* časovnih vrst y in x:

$$y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=q}^{r} \varphi_i x_{t-i} + \varepsilon_t \tag{3.9}$$

kjer je c konstanta, p red izbranega AR modela vrste y, števila od q do r pa so zamiki v vrsti x, za katere naj se test izvede. Če je katerikoli koeficient $|\varphi_i| > 0$ statistično pomemben, zavrnemo ničelno hipotezo, da x ne vpliva na y, in sprejmemo alternativno hipotezo, tj. da je vrsta x kavzalna (Granger-kavzalna) za vrsto y. Intuitivno, če lahko linearni regresijski model neke vrste bolje prilagodimo tako, da vključimo tudi pretekle člene neke druge vrste, lahko zaključimo, da je gibanje druge vrste vzročno za prvo. Dasiravno zvezo med vrstama označujemo kot kavzalno, to ne pomeni, da v resnici ena kakorkoli vpliva na stanje druge. Lahko imata obe vrsti skupen vzrok, lahko pa gre (s poljubno majhno verjetnostjo) tudi za naključje.

Grangerjevo kavzalnost uporabljamo kot orodje za določitev smiselnega seznama vrst, ki naj nastopajo skupaj v vektorskem avtoregresijskem modelu, in za določitev ustreznega reda (zamika) po enačbi:

$$\underset{i \in \mathbb{N} \cap [q,r]}{\arg\max} \, (\varphi_i \not\approx 0) \tag{3.10}$$

Poglavje 4

Razširitev programskega paketa Orange za analizo časovnih vrst

> *"All software sucks, be it open-source [or] proprietary. The only question is what can be done with particular instance of suckage, and that's where having the source matters."*
>
> Al Viro

Naš cilj je izdelava nabora orodij za interaktivno analizo, vizualizacijo in napovedovanje časovnih vrst. Ker namen diplomskega dela ni odkrivanje tople vode, smo se odločili orodja razviti v okviru ustreznega ustaljenega ogrodja in z uporabo najsodobnejših, poznanih in aktivno razvijajočih-se odprtokodnih programskih knjižnic [16, 33, 36]. Kot osnovno ogrodje smo izbrali programski paket za podatkovno rudarjenje Orange[1] [7] in sicer zaradi naslednjih očitnih prednosti:

- je prosto dostopen in odprtokoden[2] programski paket (angl. *free/libre and open-source software*), ponujen pod *"copyleft"* licenco *GNU GPLv3*;
- je znan in v panogah podatkovnega rudarjenja ter strojnega učenja uveljavljen[3][4] programski paket; v aktivnem razvoju že od leta 1996;

[1] `http://orange.biolab.si/`
[2] `https://github.com/biolab/orange3`
[3] `http://www.predictiveanalyticstoday.com/top-free-data-mining-software/`
[4] `https://www.researchgate.net/post/What_are_the_best_data_mining_tools_for_health_care_data`

- vključuje intuitivno okolje za vizualno programiranje in gradnjo podatkovnih tokov. Brez težav ga uporabljajo celo študenti družboslovja, ki ne znajo programirati;
- podpira različne strojne arhitekture in platforme; preverjeno deluje na vseh priljubljenih operacijskih sistemih za namizne računalnike, tj. na sistemih GNU/Linux, Mac OS X in celo sistemih Windows;
- je v večjem delu napisan v splošno-namenskem programskem jeziku Python, ki je zadnja leta *de facto* programski jezik na področju podatkovne znanosti.[5][6][7][8]

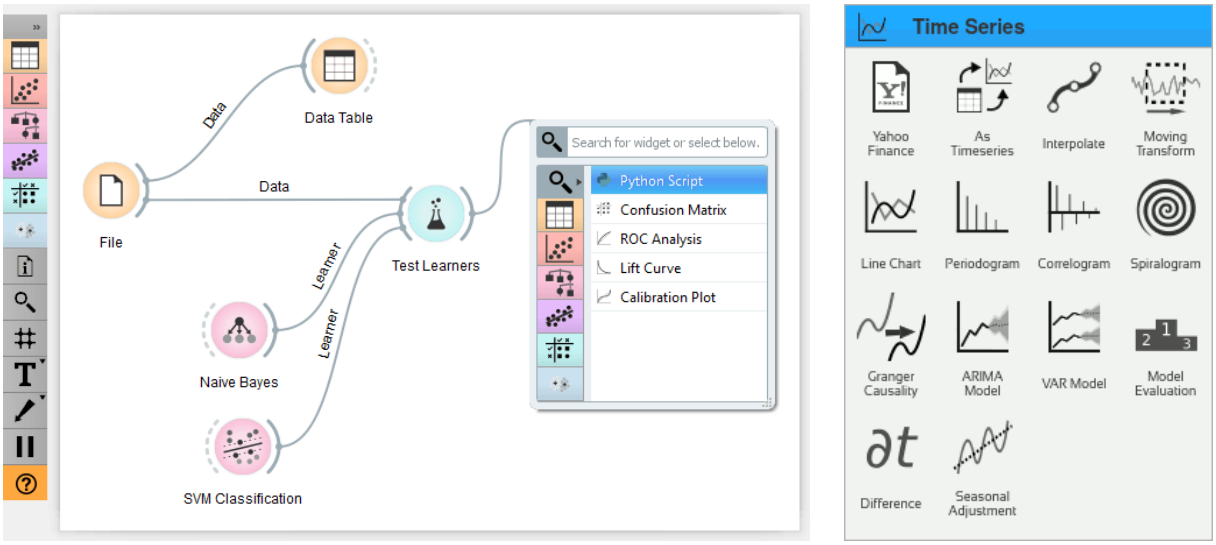
Orodje Orange je bilo uspešno uporabljeno za analizo in vizualizacijo bolnišničnih rentgenskih podatkov [19], za identifikacijo potencialno ilegalne trgovine s slonovino [13], za proučevanje genomskih podatkov [6], za analizo razvoja biofilma v vodnih distribucijskih sistemih [30], za priporočanje potencialnih genov, ki pomembno vplivajo na sposobnost bakterijske rezistence [40] itd. Prispevki, ki predstavljajo programsko orodje Orange, so bili v znanstvenih publikacijah skupaj citirani že več kot 700-krat.[9]

Največja prednost orodja Orange je zgleden in preprost uporabniški vmesnik [17]. Uporabnik lahko zgradi kompleksen delovni tok (angl. *workflow*) preprosto tako, da na delovno površino povleče ustrezne *gradnike* (angl. *widgets*) in jih poveže. Po povezavah med gradniki, od leve proti desni, teče sladek, pomarančni sok. Z dvoklikom na posamezen gradnik se odpre pogovorno okno, v katerem lahko uporabnik prilagodi dodatne možnosti gradnika.

V okviru diplomskega dela smo z razvojem ustreznega dodatka (angl. *add-on*) funkcionalnosti orodja Orange razširili tako, da je z njim mogoče učinkovito izvajati tudi analizo časovnih vrst. Slika 4.1 prikazuje grafični uporabniški vmesnik (angl. *graphical user interface, GUI*) orodja Orange z naloženo preprosto shemo z nekaj gradniki ter novo kategorijo gradnikov *Time Series*.

Skupaj smo uredili in pod odprto licenco *GNU AGPLv3+* ponudili *skoraj pet tisoč vrstic* lastne Python in JavaScript kode, preko dva tisoč besed uporabniške dokumentacije v angleščini ter štirinajst ličnih vektorskih ikon.[10] Dodatek za Orange smo kot samostojen paket z imenom Orange3-Timeseries[11] objavili tudi v indeksu Python paketov PyPI, kjer je širši javnosti na voljo za prenos. V naslednjih razdelkih so podrobneje opisane posamezne

[5] `https://www.quora.com/Why-is-Python-a-language-of-choice-for-data-scientists`
[6] `https://www.crowdflower.com/what-skills-should-data-scientists-have-in-2016/`
[7] `http://www.kdnuggets.com/2016/06/r-python-top-analytics-data-mining-data-science-software.html`
[8] `https://xkcd.com/353/`
[9] `https://scholar.google.com/scholar?as_q=orange&as_sauthors=demsar`
[10] `https://github.com/biolab/orange3-timeseries`
[11] `https://pypi.python.org/pypi/Orange3-Timeseries`



(a) (b)

Slika 4.1: (a) osnovni uporabniški vmesnik orodja Orange in (b) nova kategorija *Time Series* s štirinajst gradniki za analizo, vizualizacijo in modeliranje časovnih vrst.

uvedene spremembe, razširitve in razviti Orange gradniki, primer celotnega analitičnega delovnega toka pa je mogoče najti v poglavju 5.

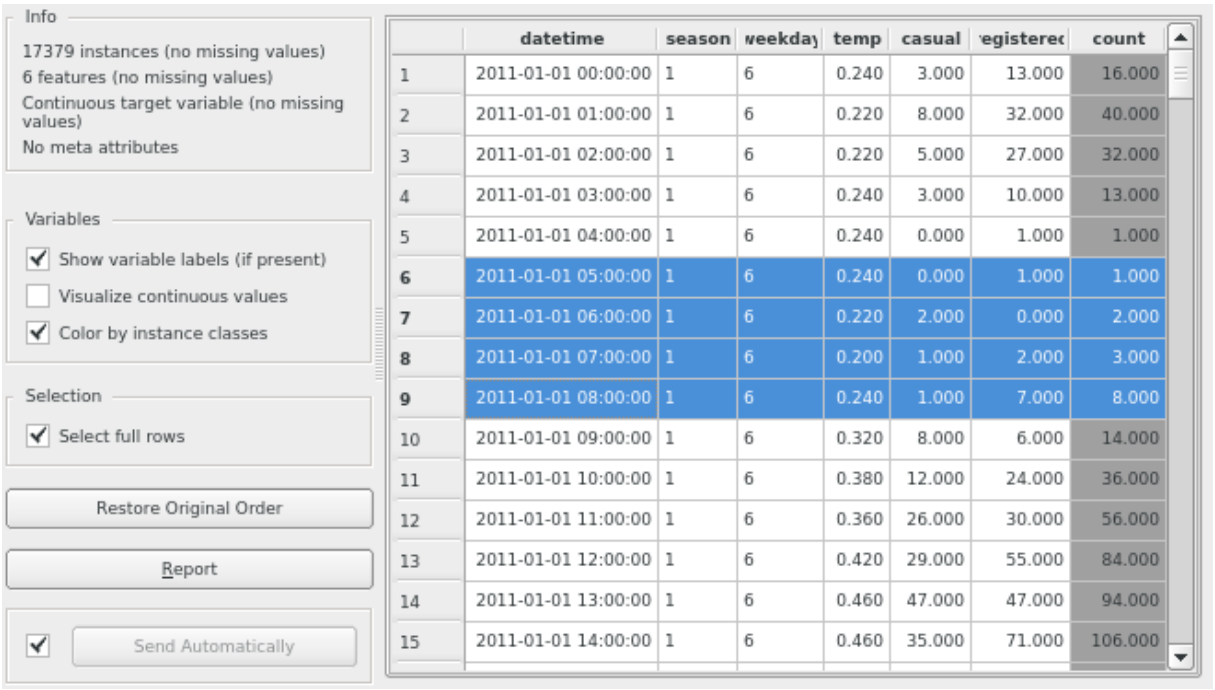
4.1 Časovna spremenljivka

Orodje Orange je tradicionalno prilagojeno za delovne tokove strojnega učenja. Osnovna podatkovna struktura je *tabela* (v Orange: `Table`), kjer imamo v vrsticah učne primere (angl. *example, instance*), v stolpcih pa njihove lastnosti (angl. *attributes, features*). Atributni prostor učnih podatkov lahko sestavljajo zvezne spremenljivke (`ContinuousVariable`) in kategorične spremenljivke (`DiscreteVariable`). Podatki, ki ne sodijo ne med ene ne med druge, so obravnavani kot nizi znakov (`StringVariable`) in jih ni mogoče uporabiti za učenje. Med take podatke sodijo tudi zapisi datuma in časa, npr. `2016-05-13 16:20:42+0200`.

Predlagana rešitev je uvedba nove časovne spremenljivke (`TimeVariable`), ki nize datuma in časa v enem od enaindvajsetih formatov ISO 8601 standarda za nedvoumen zapis časa[12] razčleni ter jih interno zapiše v obliki 64-bitnih števil s plavajočo vejico, ki predstavljajo število preteklih sekund od začetka dobe Unix (angl. *Unix epoch*) dne 1. januarja 1970 ob 00:00:00 UTC. V primeru, da zapis datuma navaja čas pred 1. januarjem 1970, je predstavitev števila ustrezno negativna.

[12] `https://en.wikipedia.org/wiki/ISO_8601`

Táko, številsko predstavitev časa je mogoče uporabiti takó v strojnem učenju kot za analizo časovnih vrst. Kakor je uporabnik vajen predstavitve časovnih vrst v programski opremi za urejanje razpredelnic (npr. LibreOffice Calc) ali skriptnih orodjih za manipulacijo podatkov (npr. Python Pandas) in odražajoč obliko večine javno-dostopnih podatkov časovnih vrst [14, 20], vrednosti posamezne časovne vrste tečejo v stolpcu (zvezna spremenljivka), morebitne nove meritve (angl. *observations*) pa dodajamo kot vrstice. Prednost te predstavitve je med drugim tudi ta, da je popolnoma združljiva z obstoječo predstavitvijo tabel v orodju Orange (slika 4.2) in je tako mogoče nad podatki, ki so časovne vrste, izvajati vse obstoječe algoritme in vizualizacije.



	datetime	season	weekday	temp	casual	registered	count
1	2011-01-01 00:00:00	1	6	0.240	3.000	13.000	16.000
2	2011-01-01 01:00:00	1	6	0.220	8.000	32.000	40.000
3	2011-01-01 02:00:00	1	6	0.220	5.000	27.000	32.000
4	2011-01-01 03:00:00	1	6	0.240	3.000	10.000	13.000
5	2011-01-01 04:00:00	1	6	0.240	0.000	1.000	1.000
6	2011-01-01 05:00:00	1	6	0.240	0.000	1.000	1.000
7	2011-01-01 06:00:00	1	6	0.220	2.000	0.000	2.000
8	2011-01-01 07:00:00	1	6	0.200	1.000	2.000	3.000
9	2011-01-01 08:00:00	1	6	0.240	1.000	7.000	8.000
10	2011-01-01 09:00:00	1	6	0.320	8.000	6.000	14.000
11	2011-01-01 10:00:00	1	6	0.380	12.000	24.000	36.000
12	2011-01-01 11:00:00	1	6	0.360	26.000	30.000	56.000
13	2011-01-01 12:00:00	1	6	0.420	29.000	55.000	84.000
14	2011-01-01 13:00:00	1	6	0.460	47.000	47.000	94.000
15	2011-01-01 14:00:00	1	6	0.460	35.000	71.000	106.000

Slika 4.2: Primer gradnika za prikaz vsebine tabele *Data Table* z naloženim delom podatkov o uporabi kolesne infrastrukture [8].

4.2 Knjižnica Highcharts

Skupaj z glavnimi dodatki Orange vsebuje preko 100 različnih gradnikov, od česar jih je več kot 30 namenjenih vizualizaciji podatkov. Žal noben od obstoječih gradnikov ni primeren za vizualizacijo časovnih vrst. Najbolj klasičen prikaz časovne vrste je črtni diagram. Robustna implementacija interaktivnega črtnega diagrama naj bi podpirala več različnih predstavitev podatkov (npr. z lomljenimi črtami ali z glajenimi, s stolpci, s svečniki (angl. *candlestick chart*)), s časovno osjo, ki omogoča posamezna interesna obdobja enostavno pobližje pogledati in ki jo je mogoče premakniti v poljubno definirano

točko v času. Zaželena je podpora za opcijsko logaritemsko ordinatno os. Več informacij o črtnem diagramu, ki smo ga razvili, je mogoče najti v razdelku 4.3.5. Upoštevši priporočila iz sekcij 2.3.3 in 3.2.1 sta potrebna prikaza za interaktivno analizo časovnih vrst vsaj še periodogram in slika avtokorelacije (več o njiju v razdelkih 4.3.6 in 4.3.7).

Zaradi izpolnjevanja vseh funkcionalnih, estetskih in interaktivnostnih zahtev, smo se po pomoč pri izgradnji teh potrebnih vizualizacij zatekli h knjižnici *Highcharts*[13]. Knjižnica Highcharts, sicer namenjena raznovrstni interaktivni vizualizaciji v spletnih brskalnikih, je izdelana v programskem jeziku JavaScript in tako ni neposredno kompatibilna s programskim jezikom Python. Razvili smo vmesno plast, ki knjižnico Highcharts omogoči v gradniku `QWebKit` ogrodja Qt, s katerim je zgrajen uporabniški vmesnik orodja Orange, in pri tem ohrani vso lepoto obstoječega programskega vmesnika.[14]

4.3 Izdelani gradniki orodja Orange

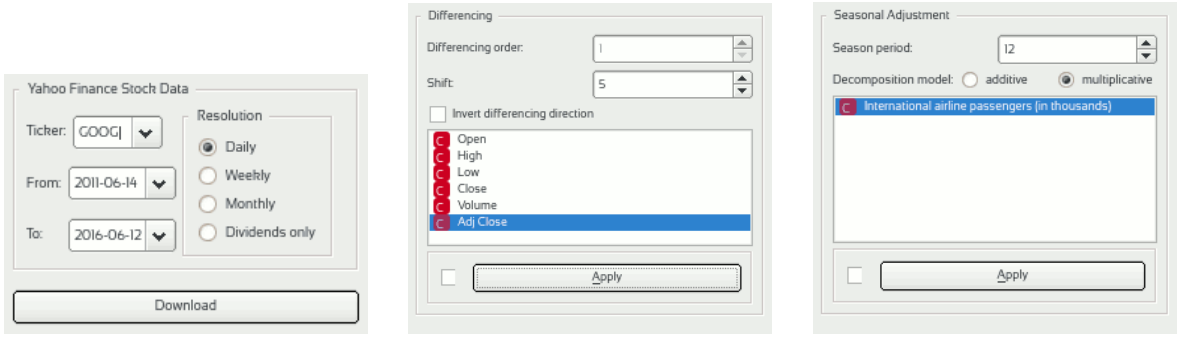
Za zadostitev cilja naloge smo v razširitvi orodja Orange pripravili štirinajst gradnikov, ki omogočijo grajenje podatkovnih tokov za analizo, vizualizacijo in napovedovanje časovnih vrst. Pomembni med njimi so podrobneje opisani v nadaljevanju.

4.3.1 Vir podatkov Yahoo Finance

Svetovne borze vrednostnih papirjev in trgi finančnih instrumentov so pogosto proučevan vir časovnih vrst [1, 11, 27]. Eden poglavitnih razlogov je najbrž ta, da so finančni trgi praktično neizčrpen vir časovnih vrst na vseh časovnih ločljivostih. Za skoraj petinštirideset tisoč delniških družb, s katerimi je mogoče trgovati na borzah po svetu,[15] je mogoče pridobiti časovne podatke na letni, mesečni, tedenski, dnevni, urni, minutni — celo na ravni vsake posamezne spremembe. Če k temu prištejmo še podatke skladov, obveznišk in valutnih trgov (angl. *forex*), pogodb na razliko v ceni (angl. *futures*) in ostalih izvedenih finančnih instrumentov, lahko zaključimo, da so finančni trgi precej ogromen vir podatkov.

Za vsako moderno orodje za analizo podatkov se spodobi, da vključuje nekaj primerov uporabe. Odločili smo se, da v razširitev orodja Orange vključimo gradnik, ki s spletne storitve Yahoo Finance[16] v izbrani ločljivosti sname zgodovinske podatke o delniški družbi ali indeksu po izbiri uporabnika. Slika 4.3a prikazuje gradnik, čigar izhod je ustrezna

[13] `http://www.highcharts.com/`
[14] `http://api.highcharts.com/`
[15] `https://www.quora.com/How-many-companies-exist-in-the-world`
[16] `https://finance.yahoo.com/`



(a) (b) (c)

Slika 4.3: Prikaz oken gradnikov: (a) za nalaganje podatkov finančnih instrumentov; (b) za odvajanje ali računanje sprememb za poljuben zamik; (c) za sezonsko prilagajanje časovnih vrst.

tabela podatkov. Primeren vir časovnih vrst v orodju Orange so tudi gospodarski in klimatski podatki Svetovne banke [39].

4.3.2 Gradnik za odvajanje

Kot omenjeno v razdelku 2.3.1, morajo analizirane časovne vrste biti stacionarne. V ta namen smo za uporabnika pripravili gradnik, kjer je mogoče časovno vrsto do dvakrat odvesti, mogoče pa je tudi izračunati razlike za več kot en zamik. Na primer, s časovno vrsto mesečnih podatkov lahko izračunamo razlike z zamikom 12, da dobimo absolutno letno spremembo. Pri tem velja opozorilo, da za del podatkov na začetku vrste, ekvivalenten izbranemu zamiku (ali redu odvajanja, če je izbran zamik enak 1), te razlike ni mogoče izračunati, zato se tiste vrednosti nadomestijo z neveljavnim številom (iz angl. *not a number, NaN*). Prikaz gradnika za odvajanje je na sliki 4.3b.

4.3.3 Sezonsko prilagajanje

Kadar se uporabnik zaveda, da določena časovna vrsta vsebuje sezonska nihanja, lahko vrsto sezonsko prilagodi z gradnikom za sezonsko prilagajanje (kot pojasnjeno v razdelku 2.2). Izbere lahko tip dekompozicije, navesti pa mora dolžino periode. Na primer, pri mesečnih podatkih in predvideni dolžini sezone eno leto je dolžina periode enaka 12. Oba tipa dekompozicije vključujeta enotno trend-cikel komponento. Prikaz okna gradnika je na sliki 4.3c, prikaz posameznih komponent vrste v črtnem diagramu pa na sliki 4.4.

4.3.4 Interpolator

Ker imajo časovne vrste lahko manjkajoče vrednosti (nekaj razlogov za pojav le-teh je moč najti v razdelku 2.3.4), ki so v orodju Orange navadno predstavljene z neveljavnimi števili (NaN), je pred uporabo funkcionalnosti ostalih gradnikov potrebno te vrednosti ustrezno nadomestiti. V interpolacijskem gradniku lahko uporabnik izbira med linearno interpolacijo, interpolacijo s kubičnimi zlepki in zamenjavo z najbližjo (prejšnjo) veljavno vrednostjo; morebitne diskretne spremenljivke, torej sekvence kategoričnih vrednosti, pa je mogoče nadomestiti z najbolj pogosto vrednostjo (modus, angl. *mode*) oz. s prejšnjo veljavno vrednostjo. Za vse navedene načine interpolacije smo uporabili algoritme iz knjižnice SciPy [16].

Omenimo, da se interpolacija izvede po principu lenega vrednotenja (angl. *lazy evaluation*), tj. vrednosti se ne interpolirajo, dokler jih nek algoritem izrecno ne potrebuje, uporabniku pa so vedno na voljo tudi izvorne, neinterpolirane vrednosti.

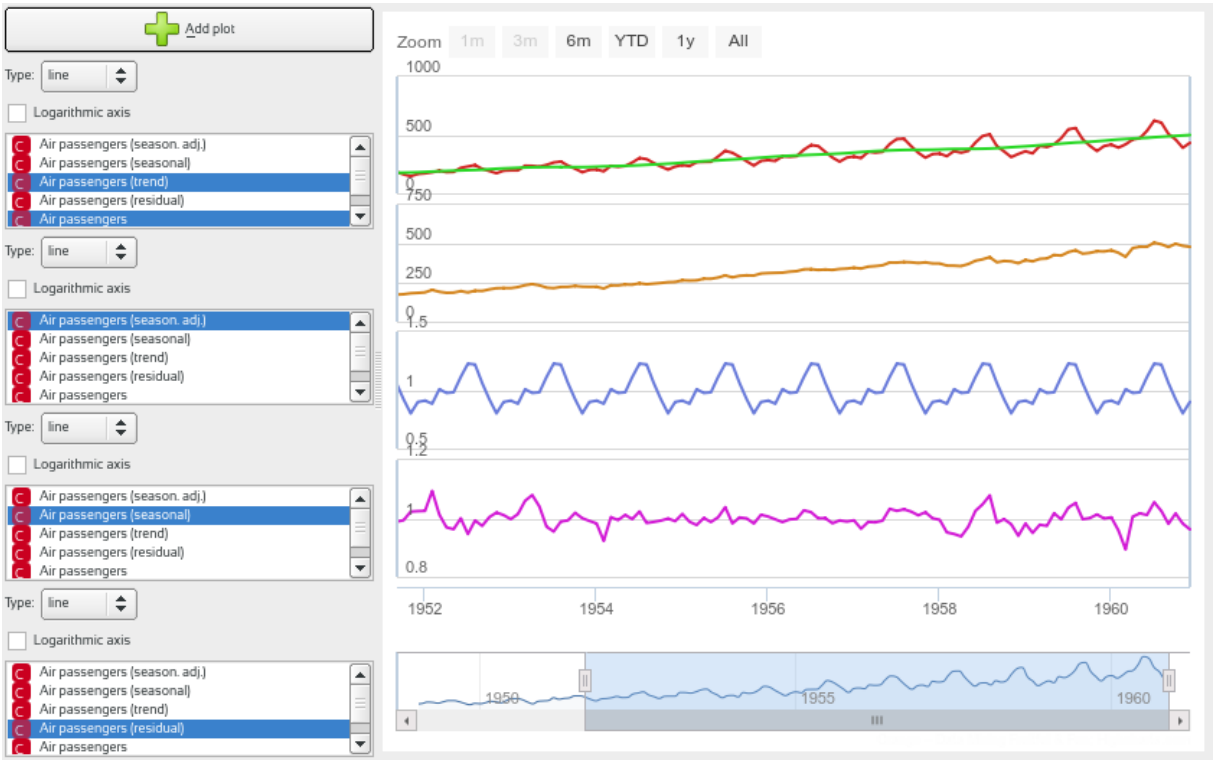
4.3.5 Črtni diagram

Človeku verjetno najbolj naravna vizualizacija časovnih vrst je črtni diagram — graf, na katerem časovna spremenljivka predstavlja abscisno os, ordinatno pa vrednosti meritev časovne vrste. Naš prispevek vključuje črtni diagram, s katerim lahko uporabnik pregleduje več časovnih vrst naenkrat, lahko nastavlja časovni razpon pregleda, tip vizualizacije (polna črta, koračna črta (angl. *step line*), stolpci, obarvana površina, zglajena polna črta) in skalo ordinatne osi (linearna ali logaritemska). Primer okna s prikazanimi petimi časovnimi vrstami je na sliki 4.4. Dodatno lahko uporabnik na vhod gradnika poveže napovedi nadaljevanja enega ali večih modelov (opisanih v razdelku 4.3.11). V tem primeru se napovedi prikažejo s črtkano črto, interval zaupanja pa je obarvan (slika 4.5).

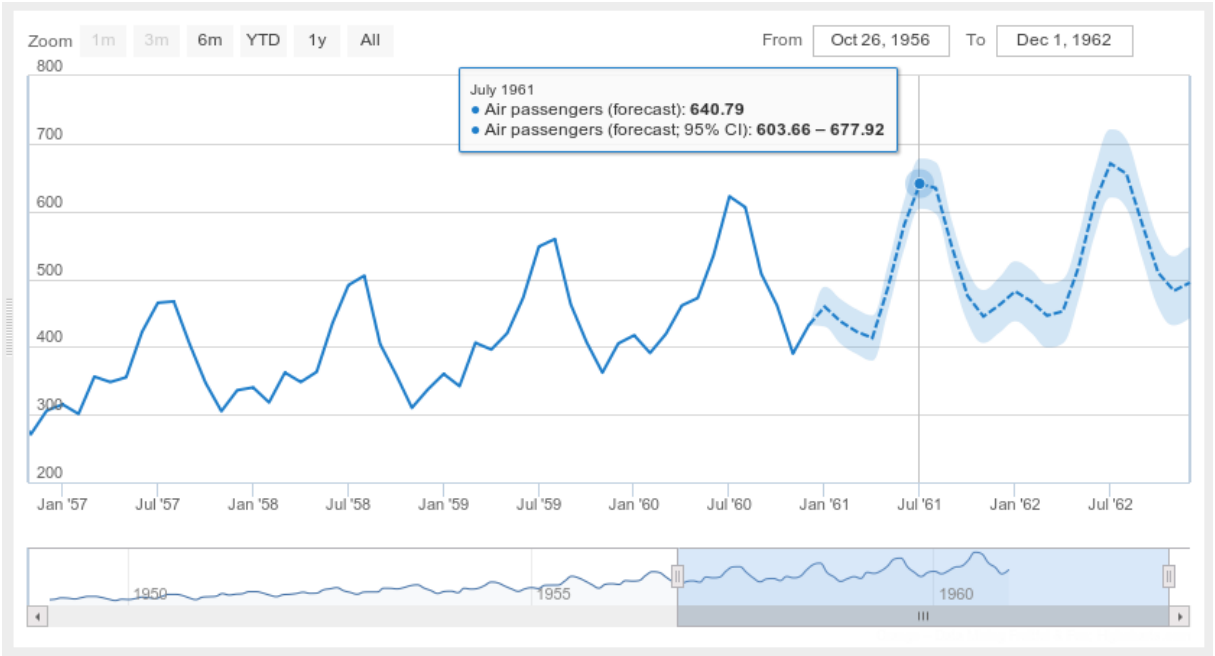
4.3.6 Korelogram

Zaporedne meritve v časovni vrsti so pogosto pozitivno korelirane: visoke vrednosti sledijo visokim in nizke nizkim. Avtokorelacijo lahko uporabimo za preverjanje, ali je časovna vrsta naključna, saj so koeficienti avtokorelacije za vse zamike naključno porajane spremenljivke nizki [27], in za iskanje najbolj pomembnih zamikov, ki jih je smiselno vključiti v avtoregresijski model. Kako za določitev zamikov uporabljamo avtokorelacijo in delno avtokorelacijo, smo povedali v razdelku 3.1.

V oknu gradnika (slika 4.6a) ima uporabnik možnost izbire med klasično in delno avtokorelacijsko funkcijo, izbere pa lahko tudi izris linij 95-odstotnega statističnega intervala



Slika 4.4: Prikaz dekompozicije (opisane v razdelku 2.2) časovne vrste letalskih potnikov v oknu gradnika za vizualizacijo črtnih diagramov. Vrste od zgoraj navzdol: originalna vrsta in trend, sezonsko prilagojena vrsta, sezonsko nihanje, inovacije (ostanek).



Slika 4.5: Prikaz časovne vrste in napovedi modela ARIMA$(8, 1, 4)$ s 95 % intervali zaupanja.

pomembnosti, ki je za avtokorelacijsko funkcijo normalnega naključnega šuma dolžine L definiran:

$$\Delta_{95\%} \approx 0 \pm \frac{1.96}{\sqrt{L}} \tag{4.1}$$

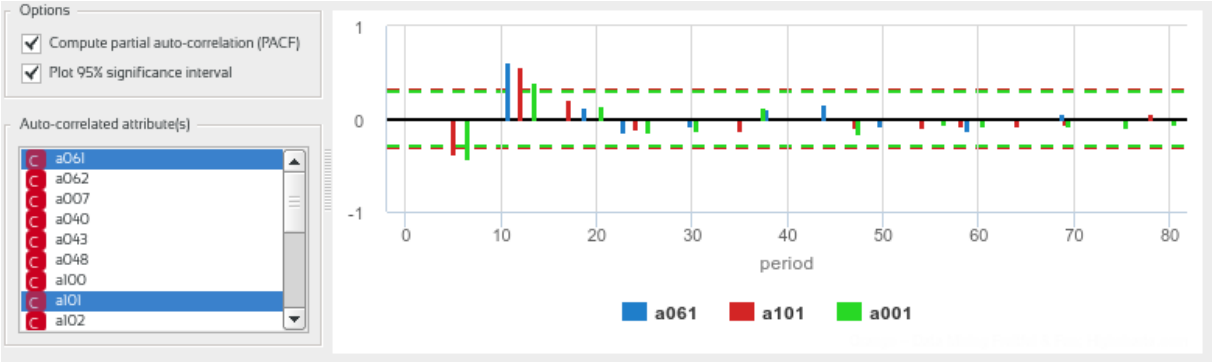
bolj natančno pa ga za zamik k izračunamo po Bartlettovi formuli[3]:

$$\Delta_{k,95\%} \approx 1.96 \cdot \sigma_k = 1.96 \cdot \sqrt{\frac{1}{L}\left(1 + 2\sum_{i=1}^{k-1} r_i^2\right)} \tag{4.2}$$

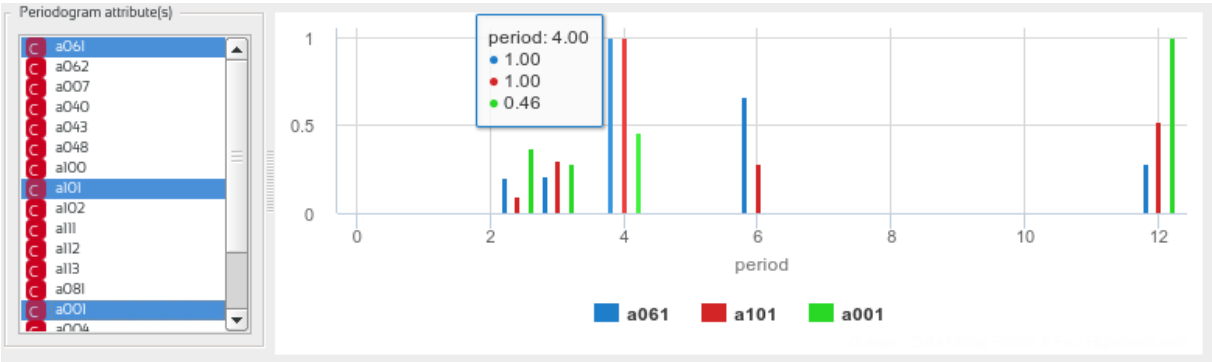
kjer je r_i koeficient avtokorelacije pri zamiku i.

4.3.7 Periodogram

V oknu gradnika Periodogram lahko uporabnik izbere časovne vrste, za katere želi prikazan periodogram. V primeru, da meritve izbrane časovne vrste niso enakomerno razmaknjene (v smislu definicije iz razdelka 2.3.2), se vrednosti periodograma namesto s FFT izračunajo z implementacijo algoritma Lomb-Scargle [22] iz knjižnice SciPy. Zaradi večje preglednosti



(a)



(b)

Slika 4.6: (a) Korelogram in (b) periodogram časovnih vrst mesečnega pretoka vozil za tri izbrane avtoceste v Franciji. Pomembne periode so dolge 4, 6 in 12 mesecev.

izris vsakokrat omejimo zgolj na množico lokalnih ekstremov.

Ker dejanske magnitude periodograma niso pomembne — odvisne so namreč od vrednosti časovne vrste, uporabnika pa za oceno pomembnih period zanima zgolj relativna velikost in ali se izrisani periodogrami večih vrst nemara ujemajo v nekaterih bistvenih periodah —, vse pomembne vrhove normiramo na interval $[0, 1]$. Primer okna s periodogramom je na sliki 4.6b.

4.3.8 Spiralogram

Naslednjo zanimivo vizualizacijo, ki smo jo razvili, imenujemo spiralogram. V principu gre za diskretizirano polarno toplotno karto (angl. *heat map*). Prednost krožne predstavitve pri podatkih časovnih vrst je, da je ta bolj naravna, saj si časovno os lahko predstavljamo kot neprekinjeno spiralo.

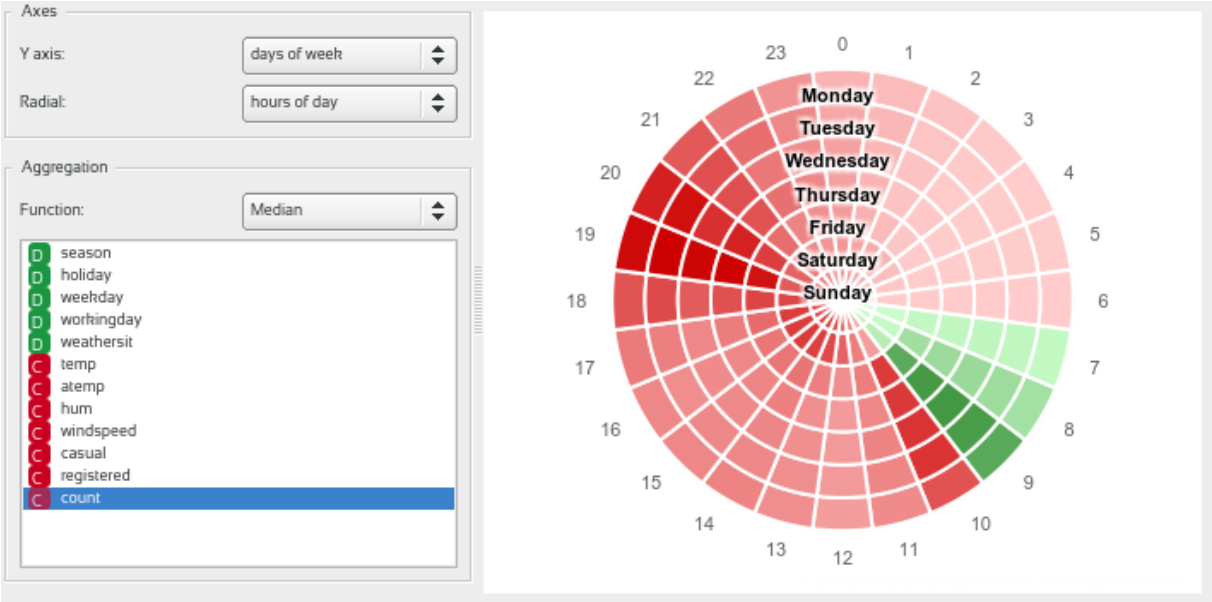
V vmesniku lahko uporabnik izbere enoti kopičenja za radialno in ordinatno os, vrste, ki jih želi prikazati, in združevalno funkcijo (angl. *aggregate function*), ki jo želi uporabiti na vsaki kopici meritev. Združevalne funkcije, ki so na voljo, so enake kot v primeru gradnika za transformiranje časovnih vrst, opisanega v razdelku 4.3.10. Primer vsebine gradnika spiralograma je na slikah 4.7 in 4.8.

4.3.9 Grangerjeva kavzalnost

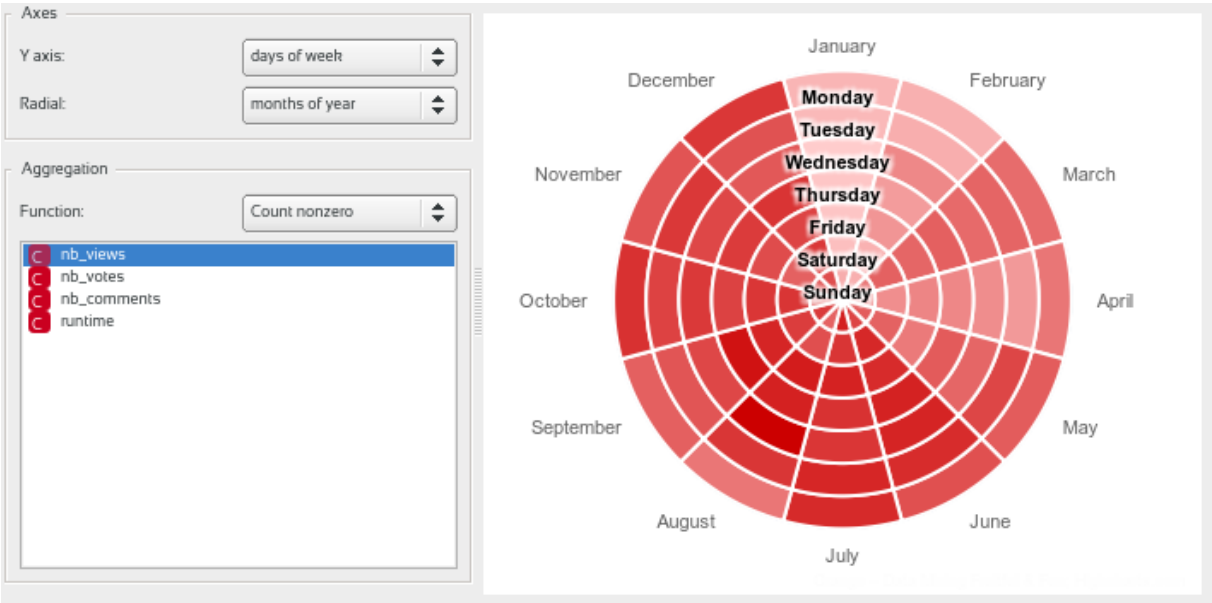
V oknu tega gradnika lahko uporabnik analizira navidez vzročno zvezo med različnimi vrstami. V vmesniku lahko določi interval zaupanja in najvišje število zamikov, ki naj jih algoritem upošteva. Uporabili smo implementacijo algoritma iz knjižnice *Statsmodels*[33], ki enačbo 3.9 izpostavi omejitvam $q = 1$ ter $p = r = m$, kjer je m od uporabnika določen največji zamik. Rezultat izračuna je razpredelnica, v kateri je razvidno, v najmanj koliko zamikih je neka vrsta Granger-kavzalna za drugo (slika 4.9).

4.3.10 Okenske transformacije

Kot smo omenili v razdelku 2.3.4, problem prisotnosti šuma v podatkih blažimo s porazdeljevanjem in z glajenjem. Med najbolj pogoste oblike glajenja sodita preprosto povprečenje prejšnjih nekaj vrednosti (angl. *simple moving average, SMA*) in eksponentno povprečenje (angl. *exponential moving average, EMA*), ki daje bolj nedavnim vrednostim večji pomen.

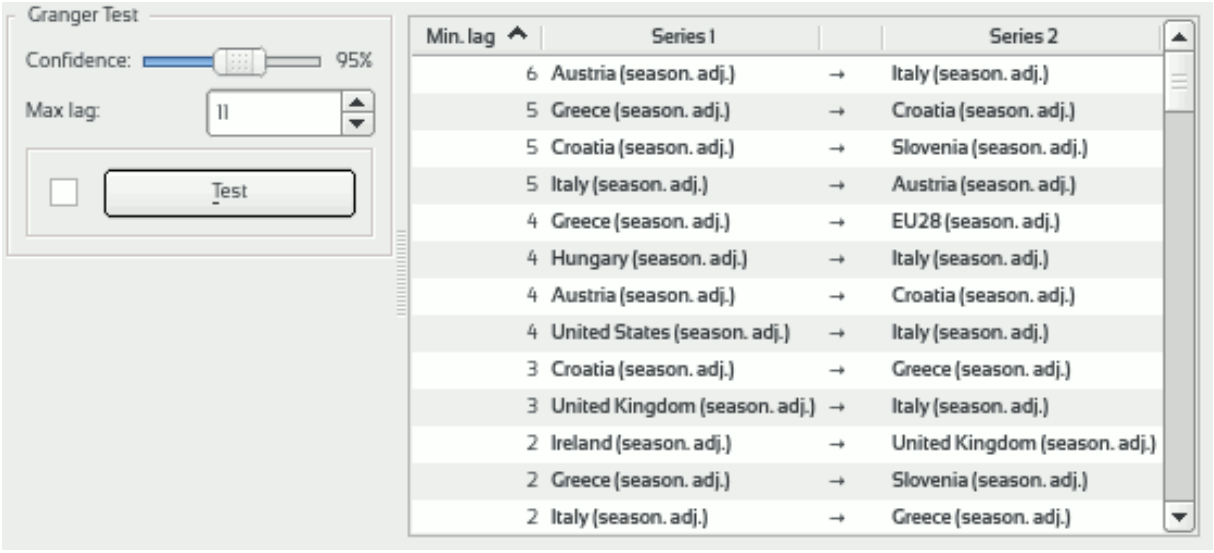


Slika 4.7: Spiralogram prikazuje mediano števila izposojenih koles po dnevih v tednu in urah v dnevu [8]; vrednosti med 7. in 10. uro so izbrane. Sklepati je mogoče, da se v centru mesta Washington, ZDA, večina kolesarjev na delo oz. "po opravkih" odpravlja med 9. in 11. uro dopoldne, domov pa se vrača med 19. in 21. uro zvečer.



Slika 4.8: Spiralogram prikazuje skupno število naloženih video vsebin na priljubljeno multimedijsko spletno stran *xHamster*[17] v obdobju od začetka leta 2008 do konca 2012 [26]. V mesecu januarju in februarju je mogoče opaziti znaten upad števila naloženih video vsebin.

[17] http://www.xhamster.com/



Min. lag	Series 1		Series 2
6	Austria (season. adj.)	→	Italy (season. adj.)
5	Greece (season. adj.)	→	Croatia (season. adj.)
5	Croatia (season. adj.)	→	Slovenia (season. adj.)
5	Italy (season. adj.)	→	Austria (season. adj.)
4	Greece (season. adj.)	→	EU28 (season. adj.)
4	Hungary (season. adj.)	→	Italy (season. adj.)
4	Austria (season. adj.)	→	Croatia (season. adj.)
4	United States (season. adj.)	→	Italy (season. adj.)
3	Croatia (season. adj.)	→	Greece (season. adj.)
3	United Kingdom (season. adj.)	→	Italy (season. adj.)
2	Ireland (season. adj.)	→	United Kingdom (season. adj.)
2	Greece (season. adj.)	→	Slovenia (season. adj.)
2	Italy (season. adj.)	→	Greece (season. adj.)

Slika 4.9: Ravni nezaposlenosti nekaterih evropskih držav so lahko Granger-kavzalne za druge.

Splošno okensko transformacijo lahko generaliziramo z naslednjo enačbo:

$$z_i = f(y_{i-w+1}, \ldots, y_i) \tag{4.3}$$

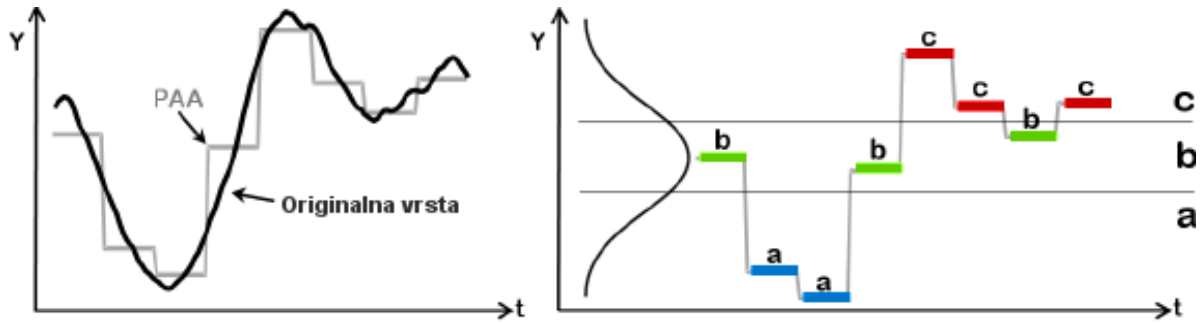
kjer so y_i vrednosti časovne vrste, $w \geq 1$ širina okna, f poljubna združevalna funkcija, z_i pa izhodne vrednosti. Vrednosti navadnega povprečja tako dobimo, če za f vzamemo:

$$f(x_1, \ldots, x_n) = \frac{\sum_{i=1}^{n} x_i}{n} \tag{4.4}$$

Na ta način z definicijo različnih funkcij in dolžin oken uvedemo različne transformacije osnovne časovne vrste.

V oknu gradnika za drseče okenske transformacije (slika 4.11) ima uporabnik možnost za vsako želeno vrsto izbrati velikost okna in združevalno funkcijo, pri čemer lahko izbira med naslednjimi: povprečje, vsota, maksimum, minimum, mediana, modus, standardni odklon, varianca, produkt, kumulativna vsota, kumulativni produkt, linearno-uteženo povprečje, eksponentno-uteženo povprečje, harmonična sredina, geometrična sredina in štetje neničelnih vrednosti. V oknu gradnika je mogoče določiti tudi, naj se okna podatkov med seboj ne prekrivajo, temveč naj se postavljajo drugo ob drugega. V tem primeru, če je izbrana funkcija povprečje, postopku rečemo *deloma nakopičena aproksimacija* (angl. *piecewise aggregate approximation, PAA*), ki predstavlja pomemben način kompresije časovnih vrst [27]. Če nadalje uporabimo gradnik za diskretizacijo (orig. *Discretize*) iz osnovnega nabora gradnikov orodja Orange, lahko dobljene vrednosti proporcionalno

diskretiziramo (angl. *equal-frequency discretization*). Strnjena predstavitev podatkov, ki je rezultat tega postopka, je znana kot *simbolna nakopičena aproksimacija* (angl. *symbolic aggregate approximation, SAX*) [21] in se uporablja kot računsko manj zahteven nadomestek DWT ali DFT pri klasifikaciji ali razvrščanju ter za iskanje pogostih sekvenc (angl. *frequent sequence mining, motif discovery*). Prikaz postopkov PAA in SAX je na sliki 4.10.

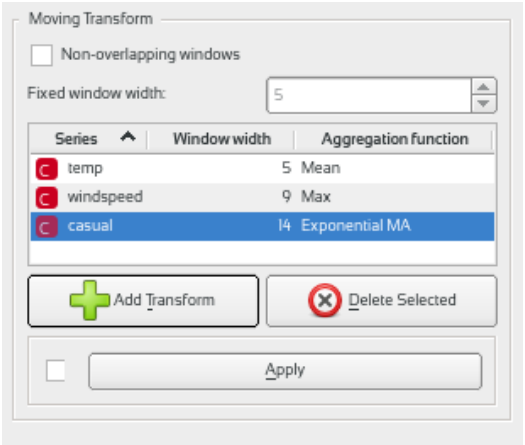


(a) (b)

Slika 4.10: Postopka (a) PAA in (b) SAX strneta originalno vrsto v niz **baabccbc**.

4.3.11 Napovedna modela VAR in ARIMA

V razvito orodje smo vključili modela VAR in ARIMA, ki smo ju podrobneje opisali v poglavju 3. Uporabili smo implementaciji, ki sta na voljo v knjižnici Statsmodels, od česar so odvisni tudi parametri, ki jih uporabnik lahko nastavlja.



Slika 4.11: Okno gradnika za drseče okenske transformacije.

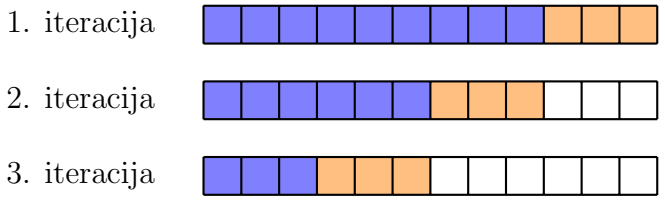
V oknu modela VAR (slika 4.13a) lahko uporabnik izbere red modela in morebitno optimizacijo po enem od navedenih informacijskih kriterijev. Določi lahko tudi, ali naj se podatkom pripne stolpec enk, kar predstavlja konstanto **c** v enačbi 3.1; stolpec linearno-naraščajočih vrednosti, kadar sumi, da je v časovni vrsti prisoten linearen trend; ali stolpec kvadratov linearno-naraščajočih vrednosti, kadar sumi, da je v časovni vrsti prisoten eksponenten trend.

V oknu modela ARIMA (slika 4.13b) pa lahko uporabnik določi število AR členov, stopnjo odvajanja ter število MA členov. Dodatno lahko izbere, ali želi preostale vrste v isti tabeli uporabiti kot zunanje, eksogene (angl. *exogenous*) vplive (model ARIMAX). V

oknih obeh modelov lahko uporabnik določi tudi, za koliko korakov v prihodnost naj bodo napovedi zgrajene in s kakšnim intervalom zaupanja. Napovedi, ki so izhod gradnikov modelov, je vključno z intervali zaupanja mogoče prikazati v črtnem diagramu (slika 4.5).

4.3.12 Vrednotenje napovednih modelov

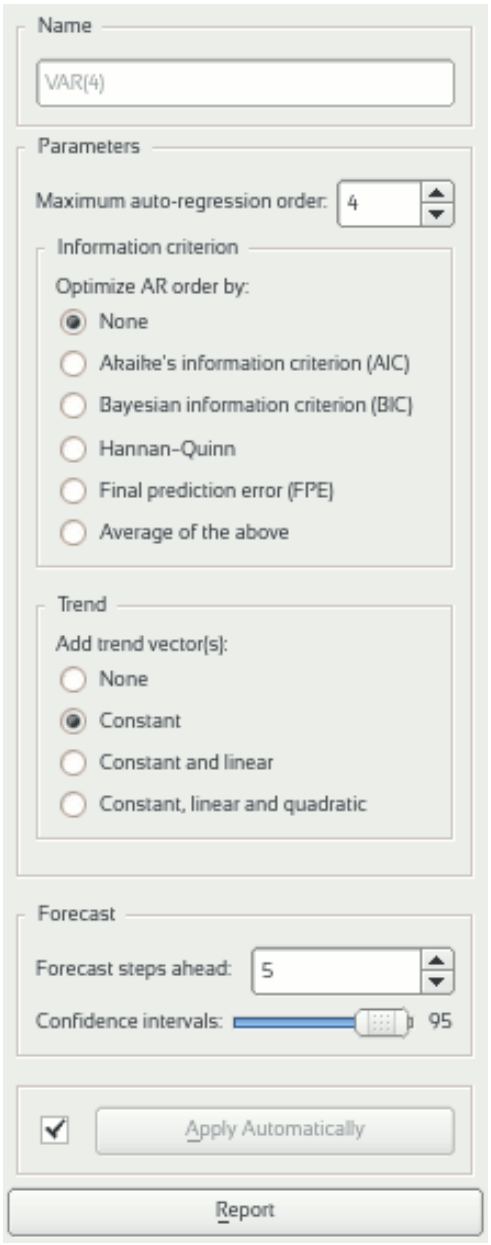
Modele, kot sta VAR in ARIMA, opisana v prejšnjem razdelku, je potrebno ovrednotiti. En način vrednotenja modelov je s pomočjo mer napak, storjenih pri prečnem preverjanju. Zasnovali smo gradnik, ki s postopkom prečnega preverjanja, podrobneje opisanega v razdelku 3.2.2, izračuna in prikaže napake, opisane v istem razdelku. Dodatno prikaže tudi vrednosti informacijskih kriterijev AIC in BIC za prilagojene modele. Uporabnik lahko nastavi število iteracij in število napovedanih prihodnjih vrednosti v vsaki iteraciji. Primer prečnega preverjanja z izbranim številom iteracij ter izbranim številom napovedi enako tri je prikazan na sliki 4.12. Okno gradnika je prikazano na sliki 4.13c.



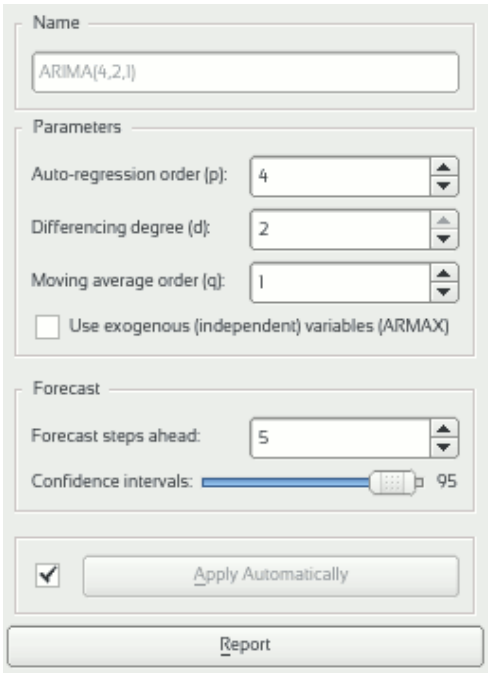
Slika 4.12: Prečno preverjanje modelov nad časovno vrsto z dvanajst vrednostmi. Modre vrednosti predstavljajo učne podatke, oranžne pa napovedi.

4.4 Programski vmesnik

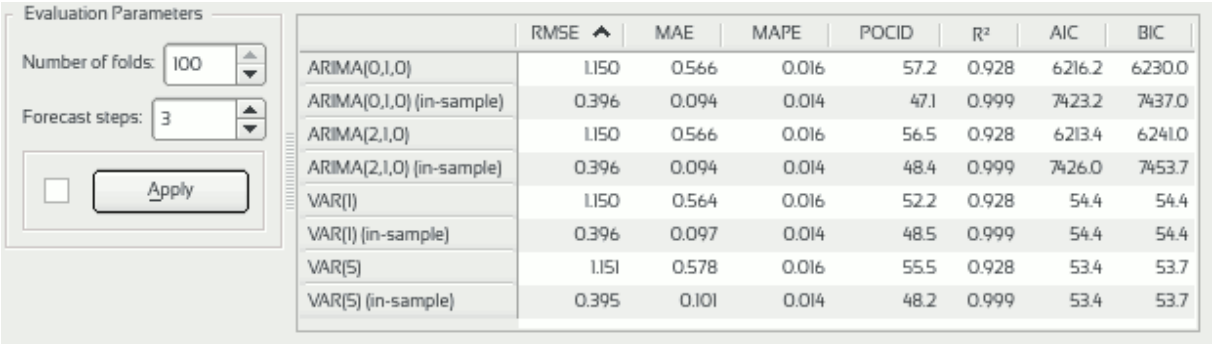
Izdelana razširitev orodja Orange vključuje tudi knjižnico z aplikacijskim programskim vmesnikom (angl. *application programming interface, API*), ki je na voljo uporabnikom, ki so vešči programiranja. Uporabniki lahko tako v svojih programih izkoriščajo večino funkcionalnosti, opisanih v razdelku 4.3. Specifično je mogoče z uporabo API klicev doseči vsaj naslednje: sneti časovne finančne podatke s spletnih storitev Yahoo Finance in Quandl; časovne vrste sezonsko prilagoditi oz. jih razstaviti na komponente; manjkajoče vrednosti v časovnih vrstah interpolirati; izračunati vrednosti avtokorelacije in delne avtokorelacije; izračunati periodogram; izračunati verjetnost Grangerjeve kavzalnosti; časovne vrste transformirati z okenskimi transformacijami; časovne vrste modelirati z modeli VAR in ARIMA, ter modele vrednotiti. Za vse naštete funkcionalnosti smo napisali tudi vso pričakovano uporabniško dokumentacijo.



(a)



(b)



	RMSE	MAE	MAPE	POCID	R^2	AIC	BIC
ARIMA(0,1,0)	1.150	0.566	0.016	57.2	0.928	6216.2	6230.0
ARIMA(0,1,0) (in-sample)	0.396	0.094	0.014	47.1	0.999	7423.2	7437.0
ARIMA(2,1,0)	1.150	0.566	0.016	56.5	0.928	6213.4	6241.0
ARIMA(2,1,0) (in-sample)	0.396	0.094	0.014	48.4	0.999	7426.0	7453.7
VAR(1)	1.150	0.564	0.016	52.2	0.928	54.4	54.4
VAR(1) (in-sample)	0.396	0.097	0.014	48.5	0.999	54.4	54.4
VAR(5)	1.151	0.578	0.016	55.5	0.928	53.4	53.7
VAR(5) (in-sample)	0.395	0.101	0.014	48.2	0.999	53.4	53.7

(c)

Slika 4.13: (a) Okno gradnika modela VAR, (b) okno gradnika modela ARIMA in (c) okno gradnika za vrednotenje modelov.

Poglavje 5

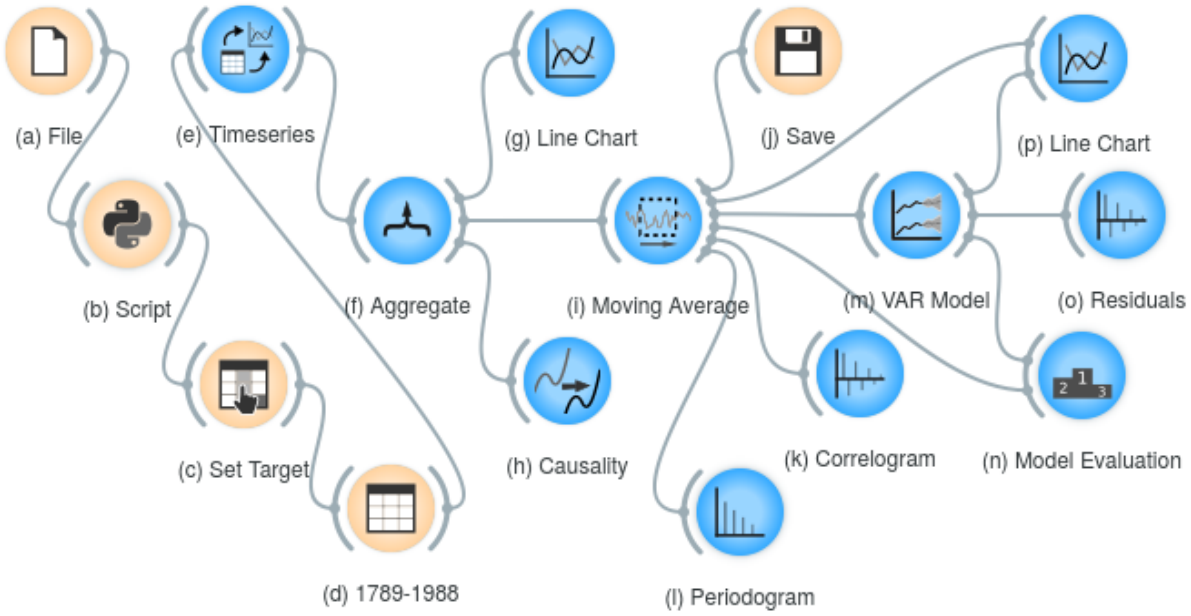
Primer uporabe: Dolgoročni temperaturni trend

"Prediction is very difficult, especially about the future."

Niels Bohr

Medvladni panel za klimatske spremembe (angl. *Intergovernmental Panel on Climate Change, IPCC*) že vrsto let opozarja, da bo en največjih problemov človeštva v tem stoletju predstavljalo globalno segrevanje (angl. *global warming*) oz. dvig povprečne letne temperature za 2 °C ali več, kar bo posledično povzročilo pogoste ekstremne vremenske razmere, izumrtje določenih rastlinskih in živalskih vrst, stalitev večine arktičnega ledu, dvig morske gladine za več metrov in premestitev stotine milijonov ljudi iz poplavljenih obmorskih predelov. Pa poglejmo, kako lahko povprečno globalno temperaturo za prihodnjih 100 let napovemo v orodju Orange z uporabo razširitve, opisane v prejšnjem poglavju tega dela.

Za zgled vzemimo delovni tok, prikazan na sliki 5.1. V gradniku (a) v Orange naložimo zbrane podatke s 171 merilnih postaj po svetu o povprečni mesečni temperaturi od leta 1700 naprej [35]. Ker Orange še nima ustreznega gradnika, v gradniku (b) s preprosto Python skripto izračunamo povprečno temperaturo preko vseh merilnih postaj za vsak mesec. To povprečno globalno temperaturo nastavimo kot napovedno spremenljivko v gradniku (c). Ker niso vse merilne postaje začele temperature meriti istočasno, v gradniku (d) izberemo samo obdobje od leta 1789 naprej, ko je k izračunu povprečne globalne temperature prispevalo vsaj trideset ločenih merilnih postaj. V gradniku (e) tabelo pretvorimo v objekt, ki predstavlja časovne vrste. V gradniku (f) časovne vrste mesečnih

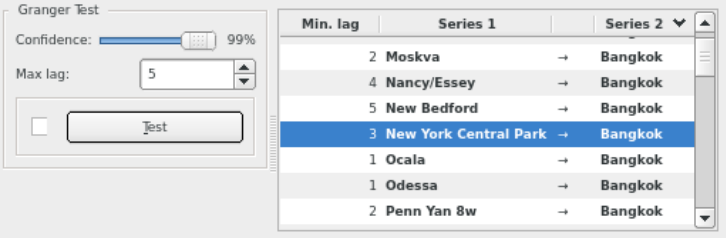


Slika 5.1: Primer delovne sheme v orodju Orange za napoved časovne vrste povprečne globalne temperature.

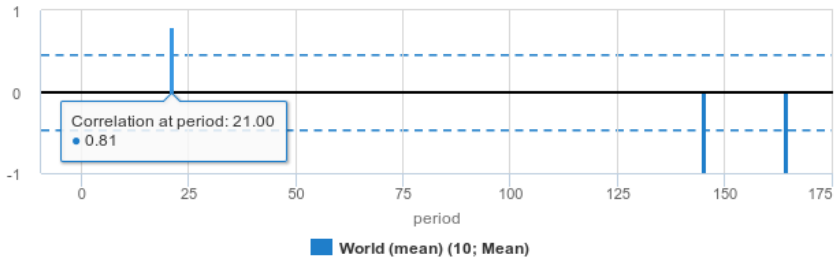
meritev v tabeli nakopičimo po letih; rezultat so časovne vrste povprečnih letnih temperatur. V oknu gradnika (h) na sliki 5.2 vidimo, da bi vrednosti določenih merilnih postaj lahko uporabili za boljše napovedi drugih. Kljub temu v modelu VAR v gradniku (m) uporabimo vse časovne vrste, da preprečimo prekomerno prileganje (angl. *overfitting*). V gradniku (i) časovno vrsto povprečne globalne temperature dodatno zgladimo z oknom velikosti $w = 10$, kar predstavlja v vsakem koraku povprečje desetih prejšnjih vrednosti (tj. temperaturno povprečje prejšnjega desetletja). Slika avtokorelacije v oknu gradnika (k) na sliki 5.3 prikazuje znatno pozitivno avtokorelacijo pri periodi 21 let, kar sovpada z dolžino solarnega cikla (cca. 11 let[1]). V gradniku (m) prilagodimo model VAR(22), ki ga v gradniku (n) ovrednotimo (slika 5.4). Povprečna absolutna napaka (MAE) modela pri prečnem preverjanju je 0,6 °C, korelogram ostankov v gradniku (o) pa nakazuje, da se model podatkom dobro prilega. V črtnem diagramu (p) pregledamo pretekle temperature in napoved za prihodnjih 100 let.

Prikaz napovedi temperatur za obdobje od leta 1988 do 2088 na sliki 5.5 je skladen s projekcijo[2] IPCC, ki predvideva maksimalen dvig temperature 1,5 do 2 °C. Kot napoveduje model, se bodo že okoli leta 2040 temperature začele spuščati, čemur bržkone pripomore predvsem popularizacija, ekonomičnost in vseprisotnost bodočih alternativnih virov energije.

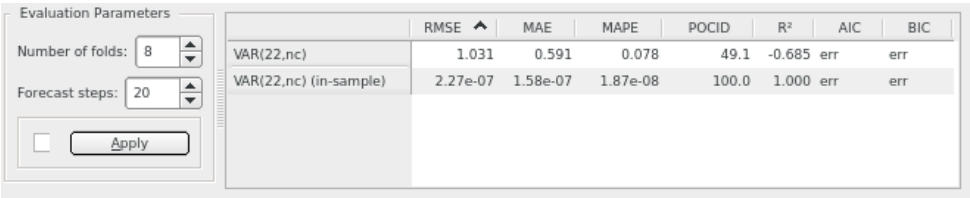
[1] https://en.wikipedia.org/wiki/Solar_cycle_21
[2] https://en.wikipedia.org/wiki/IPCC_Fifth_Assessment_Report



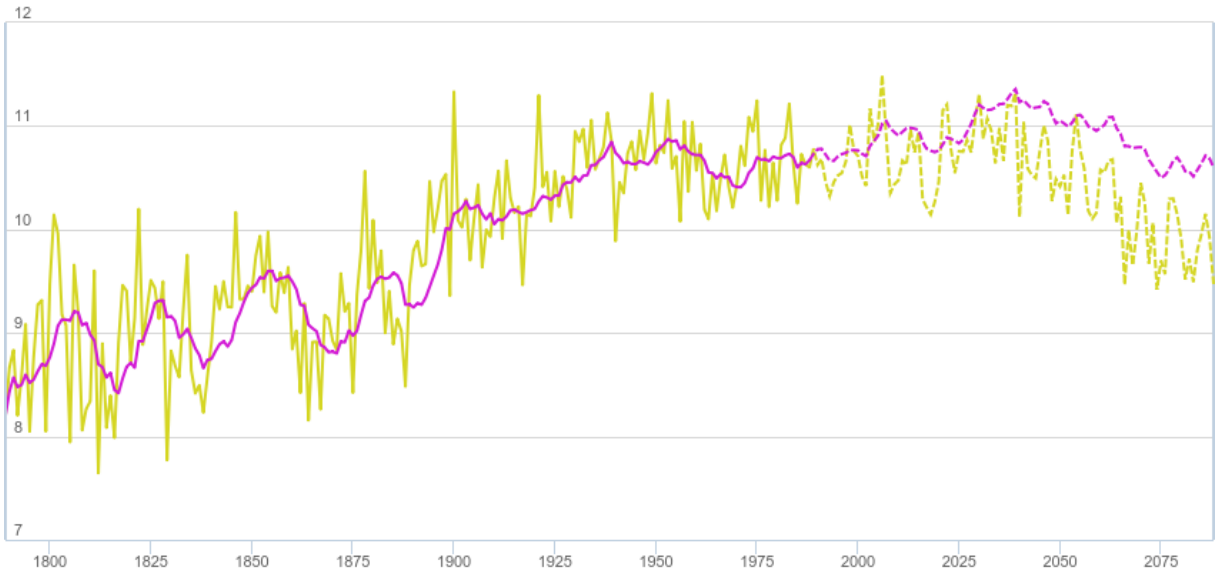
Slika 5.2: Časovne vrste določenih merilnih postaj so med seboj bolj Granger-kavzalne kot druge.



Slika 5.3: Visoka avtokorelacija pri periodi 21 let, ki je dvakratnik dolžine solarnega cikla.



Slika 5.4: Model VAR(22) pri osem-kratnem prečnem preverjanju napovedi dvajset let vnaprej naredi približno 8 % napako.



Slika 5.5: Gibanje povprečne letne temperature (rumeno) in povprečne desetletne temperature (vijolično) od leta 1789 do 1988. Črtkani črti od leta 1988 do 2088 predstavljata napovedi.

Poglavje 6

Sklepne ugotovitve

"One of my most productive days was throwing away 1000 lines of code."

Ken Thompson

V diplomskem delu smo se grobo dotaknili področja analize časovnih vrst. V obliki razširitve programskega paketa za vizualno programiranje Orange smo pripravili programsko knjižnico in štirinajst gradnikov za gradnjo podatkovnih tokov, s katerimi je mogoče časovne vrste manipulirati, transformirati, vizualizirati, analizirati, modelirati ter jim napovedovati prihodnje vrednosti. Razširitev smo objavili v ustreznem paketnem indeksu, kjer je širši javnosti ter uporabnikom orodja Orange na voljo za prenos preko vmesnika orodja Orange kot tudi samostojno.

V procesu izdelave rešitve smo naleteli na nekaj pomanjkljivosti v odprtokodnih knjižnicah Highcharts in Statsmodels, zato smo tema projektoma posredovali ustrezne popravke v obliki *zahtev za pregled sprememb* (angl. *pull request*). V primeru Statsmodels popravki iz ne-tehničnih razlogov[1] v času pisanja še niso bili integrirani.

Naša razširitev orodja Orange se je že uporabila v delu programa delavnice *Kaj nam povejo družbena omrežja?* v sklopu Poletne šole 2016 Fakultete za računalništvo in informatiko Univerze v Ljubljani. Razen nekaj specifičnih poročil o napakah, ki so vendarle pričakovani del razvoja ne-kritične programske opreme, izvajalci in udeleženci niso imeli pripomb. Glede na porast zanimanja za področja podatkovnega rudarjenja, strojnega učenja in analize časovnih vrst,[2] glede na pomanjkanje splošno-namenskih *vizualnih* orodij na tem področju in, končno, glede na število dnevnih novih uporabnikov orodja Orange upravičeno pričakujemo, da bo razširitev pogosto uporabljena.

[1] `https://github.com/statsmodels/statsmodels/pull/3031`

[2] `https://www.google.com/trends/explore#q=time+series,machine+learning`

6.1 Predlogi za nadaljevanje

V diplomskem delu smo se grobo dotaknili le nekaterih vidikov modeliranja časovnih vrst. Kot je nakazano v razdelku 3.1, so modeli, ki smo jih vključili, uporabni le za napovedovanje prihodnjih vrednosti, nič pa se ne posvečamo klasifikaciji ali razvrščanju časovnih vrst. S tem bi se srečali, na primer, ko bi želeli na podlagi gibanja bruto domačega proizvoda (angl. *gross domestic product, GDP*) neke države, ali drugega primernega ekonomskega indikatorja, državo opredeliti med visoko-razvite, srednje-razvite oz. manj-razvite države ali določiti, katerim drugim državam je ta najbolj podobna ali ko bi želeli za EEG sliko nekega človeka izraziti, ali ima človek srčno napako oz. druge težave v krvožilnem sistemu, jih je imel v preteklosti ali pa jih še ni imel. Za klasifikacijo časovnih vrst ter za razvrščanje je bistvena definicija razdalje med dvema vrstama. Orodje Orange, ki smo ga razširili, določene mere razdalje (npr. evklidsko) že vsebuje, potrebno pa bi bilo implementirati vsaj tudi razdaljo DTW. V primeru časovnih sekvenc (zaporednih vrednosti kategoričnih spremenljivk) je smiselna mera razdalje tudi *najdaljše skupno podzaporedje* (angl. *longest common subsequence, LCS*, tudi *LCSS*). Za klasifikacijo in razvrščanje časovnih vrst bi bilo potrebno v Orange tudi uvesti nek splošen gradnik za *transponiranje* tabele, saj časovne vrste po zdajšnji implementaciji tečejo v stolpcih (razdelek 4.1), v orodju Orange pa razrede oz. skupine pripisujemo primerom v vrsticah. V primeru klasifikacije ali razvrščanja ogromnih časovnih vrst, bi bilo menda smotrno razmisliti tudi o uvedbi raznih vrst *kompresije* časovnih vrst, npr. z obrezovanjem (angl. *cliping*), kjer vrednosti večje od povprečja nadomestimo z 1, vrednosti manjše pa z 0, ali pa s pomočjo valčne transformacije DWT oz. z DFT, kar pohitri nadaljnje računanje DTW [27].

Vredno bi bilo tudi pogledati, ali je mogoče v Orange vključiti še kakšne popularne ekonometrijske modele, kot so VARMAX, SARIMA, GARCH in GAS. Zanimivo bi bilo pogledati, ali lahko z multiresolucijskimi modeli konsistentno dosegamo bistveno boljše napovedi. Če Orange nekoč postane orodje, primerno za analizo velikega podatkovja (angl. *big data*), bi bilo smiselno algoritme za analizo časovnih vrst prilagoditi tako, da bodo ti podpirali pretočne podatke (angl. *streaming*) in sprotno učenje (angl. *online algorithms*). Nenazadnje bi bilo potrebno implementirati tudi vsaj kakšen algoritem, ki operira nad nezveznimi časovnimi sekvencami, kot je predstavitev SAX iz razdelka 4.3.10, npr. iskanje najbolj pogostih sekvenc z algoritmom BIDE [37]. Ti odprti problemi so lahko tema neke druge, podobne naloge.

Literatura

[1] T. G. Andersen in sod. *Handbook of Financial Time Series*. Springer Science & Business Media, 2009.

[2] C. Bergmeir, R. J. Hyndman in B. Koo. *A Note on the Validity of Cross-Validation for Evaluating Time Series Prediction*. Monash Econometrics and Business Statistics Working Papers 10/15. Monash University, Department of Econometrics in Business Statistics, okt. 2015.

[3] G. E. Box in G. M. Jenkins. *Time series analysis: forecasting and control*. Holden-Day, 1976.

[4] P. J. Brockwell in R. A. Davis. *Introduction to time series and forecasting*. Springer Science & Business Media, 2006.

[5] P. Cortez in sod. "Internet traffic forecasting using neural networks". V: *Neural Networks, 2006. IJCNN'06. International Joint Conference on*. IEEE. 2006, str. 2635–2642.

[6] T. Curk in sod. "Microarray data mining with visual programming". V: *Bioinformatics* 21.3 (2005), str. 396–398.

[7] J. Demšar in sod. "Orange: Data Mining Toolbox in Python". V: *Journal of Machine Learning Research* 14 (2013), str. 2349–2353.

[8] H. Fanaee-T in J. Gama. "Event labeling combining ensemble detectors and background knowledge". V: *Progress in Artificial Intelligence* (2013), str. 1–15.

[9] J. Gama. *Knowledge Discovery from Data Streams*. 1st. Chapman & Hall/CRC, 2010.

[10] C. W. Granger. "Investigating causal relations by econometric models and cross-spectral methods". V: *Econometrica: Journal of the Econometric Society* (1969), str. 424–438.

[11] X. Han in sod. "A Novel Time Series Forecasting Approach with Multi-Level Data Decomposing and Modeling". V: *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*. Zv. 1. IEEE. 2006, str. 1712–1716.

[12] E. J. Hannan in J. Rissanen. "Recursive estimation of mixed autoregressive-moving average order". V: *Biometrika* 69.1 (1982), str. 81–94.

[13] J. Hernandez-Castro in D. L. Roberts. "Automatic detection of potentially illegal online sales of elephant ivory via data mining". V: *PeerJ Computer Science* 1 (2015), e10.

[14] R. J. Hyndman. *Time Series Data Library*. URL: `http://data.is/TSDLdemo`.

[15] R. J. Hyndman in G. Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2014.

[16] E. Jones, T. Oliphant, P. Peterson in sod. *SciPy: Open source scientific tools for Python*. 2001–.

[17] J. Kernc. "Orodje za interaktivno analizo časovnih vrst". Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, 2016.

[18] D. Kifer, S. Ben-David in J. Gehrke. "Detecting change in data streams". V: *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment. 2004, str. 180–191.

[19] W. Kovacs in sod. "Retrieval, visualization, and mining of large radiation dosage data". V: *Information Retrieval Journal* (2016), str. 1–21.

[20] M. Lichman. *UCI Machine Learning Repository*. 2013.

[21] J. Lin in sod. "A symbolic representation of time series, with implications for streaming algorithms". V: *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*. ACM. 2003, str. 2–11.

[22] N. R. Lomb. "Least-squares frequency analysis of unequally spaced data". V: *Astrophysics and space science* 39.2 (1976), str. 447–462.

[23] H. Lütkepohl. *New Introduction to Multiple Time Series Analysis*. Springer Berlin Heidelberg, 2007.

[24] D. Machiwal in M. K. Jha. *Hydrologic time series analysis: theory and practice*. Springer Science & Business Media, 2012.

[25] L. Mahadeva in P. Robinson. *Unit root testing to help model building*. Centre for Central Banking Studies, Bank of England, 2004.

[26] A. Mazieres in sod. "Deep tags: toward a quantitative analysis of online pornography". V: *Porn Studies* 1.1-2 (2014), str. 80–95.

[27] T. Mitsa. *Temporal Data Mining*. 1st. Chapman & Hall/CRC, 2010.

[28] R. Nau. *Statistical forecasting: notes on regression and time series analysis*. Fuqua School of Business, Duke University. 2015.

[29] *NIST/SEMATECH e-handbook of statistical methods*. 2016. URL: `http://www.itl.nist.gov/div898/handbook/`.

[30] E. Ramos-Martínez in sod. "Pre-processing and visualization of biofilm development in drinking water distribution systems". V: (2014).

[31] K. Rehfeld in sod. "Comparison of correlation analysis techniques for irregularly sampled time series". V: *Nonlinear Processes in Geophysics* 18.3 (2011), str. 389–404.

[32] A. Schuster. "On the periodicities of sunspots". V: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 206 (1906), str. 69–100.

[33] S. Seabold in J. Perktold. "Statsmodels: Econometric and statistical modeling with Python". V: *9th Python in Science Conference*. 2010.

[34] C. A. Sims. "Macroeconomics and reality". V: *Econometrica: Journal of the Econometric Society* (1980), str. 1–48.

[35] R. S. Vose in sod. *The Global Historical Climatology Network: Long-term monthly temperature, precipitation, sea level pressure, and station pressure data*. Zv. 3912. Oak Ridge National Laboratory Oak Ridge, Tennessee, 1992.

[36] S. van der Walt, S. C. Colbert in G. Varoquaux. "The NumPy Array: A Structure for Efficient Numerical Computation". V: *Computing in Science Engineering* 13.2 (mar. 2011), str. 22–30.

[37] J. Wang, J. Han in C. Li. "Frequent Closed Sequence Mining without Candidate Maintenance". V: *IEEE Transactions on Knowledge and Data Engineering* 19.8 (avg. 2007), str. 1042–1056.

[38] X. Xi in sod. "Fast time series classification using numerosity reduction". V: *Proceedings of the 23rd international conference on Machine learning*. ACM. 2006, str. 1033–1040.

[39] M. Zidar. "Dostop do gospodarskih, klimatskih in drugih podatkov Svetovne banke v orodju Orange". Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, 2016.

[40] M. Žitnik in sod. "Gene Prioritization by Compressive Data Fusion and Chaining". V: *PLoS Comput Biol* 11.10 (okt. 2015), str. 1–18.

[41] E. Zivot in J. Wang. "Vector Autoregressive Models for Multivariate Time Series". V: *Modeling Financial Time Series with S-PLUS®*. New York, NY: Springer New York, 2006, str. 385–429.